

112740-520 **IC16 Rec'd PCT/PTO** SEP 10 2001

## TRANSMITTAL LETTER TO THE UNITED STATES

DESIGNATED/ELECTED OFFICE (DO/EO/US)

CONCERNING A FILING UNDER 35 U.S.C. 371

U.S. APPLICATION NO. (IF KNOWN, SEE 37 CFR

**09/936544**INTERNATIONAL APPLICATION NO.  
**PCT/EP00/02070**INTERNATIONAL FILING DATE  
**09 March 2000**PRIORITY DATE CLAIMED  
**01 March 1999**

TITLE OF INVENTION

**METHOD AND NETWORK ELEMENT FOR OPERATING A TELECOMMUNICATIONS NETWORK**

APPLICANT(S) FOR DO/EO/US

**Maria Froeschl et al.**

Applicant herewith submits to the United States Designated/Elected Office (DO/EO/US) the following items and other information:

1. ☒ This is a **FIRST** submission of items concerning a filing under 35 U.S.C. 371.
2. ☐ This is a **SECOND** or **SUBSEQUENT** submission of items concerning a filing under 35 U.S.C. 371.
3. ☒ This is an express request to begin national examination procedures (35 U.S.C. 371(f)) at any time rather than delay examination until the expiration of the applicable time limit set in 35 U.S.C. 371(b) and PCT Articles 22 and 39(1).
4. ☒ A proper Demand for International Preliminary Examination was made by the 19th month from the earliest claimed priority date.
5. ☒ A copy of the International Application as filed (35 U.S.C. 371 (c) (2))
  - a. ☒ is transmitted herewith (required only if not transmitted by the International Bureau).
  - b. ☐ has been transmitted by the International Bureau.
  - c. ☐ is not required, as the application was filed in the United States Receiving Office (RO/US).
6. ☒ A translation of the International Application into English (35 U.S.C. 371(c)(2)).
7. ☒ A copy of the International Search Report (PCT/ISA/210).
8. ☒ Amendments to the claims of the International Application under PCT Article 19 (35 U.S.C. 371 (c)(3))
  - a. ☒ are transmitted herewith (required only if not transmitted by the International Bureau).
  - b. ☐ have been transmitted by the International Bureau.
  - c. ☐ have not been made; however, the time limit for making such amendments has NOT expired.
  - d. ☐ have not been made and will not be made.
9. ☒ A translation of the amendments to the claims under PCT Article 19 (35 U.S.C. 371(c)(3)).
10. ☐ An oath or declaration of the inventor(s) (35 U.S.C. 371 (c)(4)).
11. ☒ A copy of the International Preliminary Examination Report (PCT/IPEA/409).
12. ☐ A translation of the annexes to the International Preliminary Examination Report under PCT Article 36 (35 U.S.C. 371 (c)(5)).

**Items 13 to 20 below concern document(s) or information included:**

13. ☒ An Information Disclosure Statement under 37 CFR 1.97 and 1.98.
14. ☐ An assignment document for recording. A separate cover sheet in compliance with 37 CFR 3.28 and 3.31 is included.
15. ☒ A **FIRST** preliminary amendment.
16. ☐ A **SECOND** or **SUBSEQUENT** preliminary amendment.
17. ☒ A substitute specification.
18. ☐ A change of power of attorney and/or address letter.
19. ☒ Certificate of Mailing by Express Mail
20. ☒ Other items or information:

**Submission of Drawings - Figures 1-5 on five sheets**

09/936544

PCT/EP00/02070

112740-320

21. The following fees are submitted:

CALCULATIONS PTO USE ONLY

**BASIC NATIONAL FEE (37 CFR 1.492 (a) (1) - (5)) :**

- ☐ Neither international preliminary examination fee (37 CFR 1.482) nor international search fee (37 CFR 1.445(a)(2)) paid to USPTO and International Search Report not prepared by the EPO or JPO ..... \$1,000.00
- ☒ International preliminary examination fee (37 CFR 1.482) not paid to USPTO but International Search Report prepared by the EPO or JPO ..... \$860.00
- ☐ International preliminary examination fee (37 CFR 1.482) not paid to USPTO but international search fee (37 CFR 1.445(a)(2)) paid to USPTO ..... \$710.00
- ☐ International preliminary examination fee paid to USPTO (37 CFR 1.482) but all claims did not satisfy provisions of PCT Article 33(1)-(4) ..... \$690.00
- ☐ International preliminary examination fee paid to USPTO (37 CFR 1.482) and all claims satisfied provisions of PCT Article 33(1)-(4) ..... \$100.00

**ENTER APPROPRIATE BASIC FEE AMOUNT =****\$860.00**

Surcharge of **\$130.00** for furnishing the oath or declaration later than ☐ 20 ☐ 30 months from the earliest claimed priority date (37 CFR 1.492 (e)).

**\$0.00**

CLAIMS	NUMBER FILED	NUMBER EXTRA	RATE
Total claims	17 - 20 =	0	x \$18.00
Independent claims	3 - 3 =	0	x \$80.00
Multiple Dependent Claims (check if applicable).			<input type="checkbox"/>

**\$0.00****\$0.00****\$0.00****TOTAL OF ABOVE CALCULATIONS =****\$860.00**

Reduction of 1/2 for filing by small entity, if applicable. Verified Small Entity Statement must also be filed (Note 37 CFR 1.9, 1.27, 1.28) (check if applicable).

☐**\$0.00****SUBTOTAL =****\$860.00**

Processing fee of **\$130.00** for furnishing the English translation later than ☐ 20 ☐ 30 months from the earliest claimed priority date (37 CFR 1.492 (f)).

+

**\$0.00****TOTAL NATIONAL FEE =****\$860.00**

Fee for recording the enclosed assignment (37 CFR 1.21(h)). The assignment must be accompanied by an appropriate cover sheet (37 CFR 3.28, 3.31) (check if applicable).

☐**\$0.00****TOTAL FEES ENCLOSED =****\$860.00**

Amount to be:

refunded

\$

charged

\$

☒ A check in the amount of **\$860.00** to cover the above fees is enclosed.

☐ Please charge my Deposit Account No. \_\_\_\_\_ in the amount of \_\_\_\_\_ to cover the above fees.  
A duplicate copy of this sheet is enclosed.

☒ The Commissioner is hereby authorized to charge any fees which may be required, or credit any overpayment to Deposit Account No. **02-1818** A duplicate copy of this sheet is enclosed.

**NOTE:** Where an appropriate time limit under 37 CFR 1.494 or 1.495 has not been met, a petition to revive (37 CFR 1.137(a) or (b)) must be filed and granted to restore the application to pending status.

SEND ALL CORRESPONDENCE TO:

William E. Vaughan (Reg. No. 39,056)  
Bell, Boyd & Lloyd LLC  
P.O. Box 1135  
Chicago, Illinois 60690

SIGNATURE

William E. Vaughan

NAME

39,056

REGISTRATION NUMBER

September 10, 2001

DATE

BOX PCT

IN THE UNITED STATES ELECTED/DESIGNATED OFFICE  
OF THE UNITED STATES PATENT AND TRADEMARK OFFICE  
UNDER THE PATENT COOPERATION TREATY-CHAPTER II

5

**PRELIMINARY AMENDMENT**

APPLICANTS: Maria Froeschl et al. DOCKET NO: 112740-320

SERIAL NO: GROUP ART UNIT:

10

EXAMINER:

INTERNATIONAL APPLICATION NO: PCT/EP00/02070

INTERNATIONAL FILING DATE: 09 March 2000

INVENTION: METHOD AND NETWORK ELEMENT FOR  
OPERATING A TELECOMMUNICATIONS NETWORK

15

Assistant Commissioner for Patents,  
Washington, D.C. 20231

Sir:

20

Please amend the above-identified International Application before entry  
into the National stage before the U.S. Patent and Trademark Office under  
U.S.C. §371 as follows:

**In the Specification:**

Please replace the Specification of the present application, including the  
Abstract, with the following Substitute Specification:

25

**SPECIFICATION**

**TITLE OF THE INVENTION**

**METHOD AND NETWORK ELEMENT FOR  
OPERATING A TELECOMMUNICATIONS NETWORK**

30

**BACKGROUND OF THE INVENTION**

The present invention relates to a method for operating a  
telecommunications network, in which a network element at a network node of the  
telecommunications network is controlled by a control computer. The network  
element is, for example, a switching office for switching links, what is referred to  
as a cross-connector or a concentrator unit for connecting a number of subscribers

35

09/936544-0340

to a connecting line. In addition to the operating system for operating the control computer, a number of application programs, during whose execution application objects are processed, are stored in the control computer. The application objects are associated with data with a predefined data structure and preferably also predefined methods for processing the data. The data structure and the methods are dependent on a class which is also to be specified when the respective application object is generated. A link via which the control computer is maintained via maintenance messages is set up between a service computer and a control computer.

Such methods are used for controlling the telecommunications network if, for example, a new switching device is to be put into operation as a network element or if subscriber data in the switching device have to be changed at a later date, such as is the case when new subscribers are connected or when an existing subscriber moves. Efficient methods for controlling the telecommunications network are obtained if what are referred to as open systems are used which are programmed in compliance with standards which apply worldwide. For example, standards of the ISO (International Standardization Organization) and of the ITU (International Telecommunication Union) with its body the ITU-T, known earlier as the CCITT (International Telegraph and Telephone Consultative Committee) relate to the setting up of such open systems. A separate control network is to be used to control the telecommunications network. The interfaces between the service computer and switching device are standardized in protocols Q1, Q2 and Q3.

The application objects are defined as objects of an object-oriented language, for example in the language C++ or CHILL. If the application programs are developed, it is necessary to ensure that the control network also operates without faults with the new application programs. As such, in particular, application objects which are considered by the service computer as belonging to an original class cannot readily be assigned to an amended alternative class.

This problem is mentioned in the CCITT standard X.720 (01/92) - "Information Technology - Open Systems Interconnection - Structure of



Management Information: Management Information Model” - in the section 5.2.1.  
In the section 5.2.3 of the standard X.720, two methods for solving the problem are  
proposed. In the first method, on the part of the application program, a  
programming technique is used which takes allomorphy into consideration.

- 5 Allomorphy is the ability of a specific application object of the alternative class to  
be controlled as if it were an object of the original class if this ability arises as a  
result of measures on the part of the application program. The other method  
includes taking measures on the part of the service computer which permit  
cooperation between the service computer and the application program even when  
10 the application program is developed.

- European patent application EP-A-0 817 422 discloses a method for  
implementing controlled objects in a subsystem of a controlled system in a  
network, at least one control system and one controlled system being present. The  
controlled objects are implemented independently of other subsystems without  
15 knowing the type of the objects in the other subsystems. They can be connected to  
other objects and can transmit messages to them. For this purpose, a first object is  
generated for cooperating with an abstract object. The abstract object has a defined  
interface which is called up using the first object and which inherits the abstract  
object from a second object, the generated second object being unknown to the  
20 first, and the second object being intended for cooperation with the first object. The  
first object which cooperates with the second object considers the second object as  
an object of the abstract type. However, in this method known from  
EP-A-0 817 422, there is no use of programming technology which takes into  
consideration the allomorphy in accordance with the section 5.2.3 of the standard  
25 X.720.

An object of the present invention, therefore, is to provide a simple method  
for operating a telecommunications network in which allomorphy is taken into  
consideration.

#### SUMMARY OF THE INVENTION

- 30 In the method according to the present invention, in each case a class  
identifier in which the class to which the maintenance message relates is specified

is determined from the maintenance messages during the processing of an interface program which is used for a number of application programs. The class identifier in the maintenance message specifies the class, known in the service computer, of an application object to be processed. As a result of the development, the class  
5 known in the service computer sometimes deviates from the actual class of the application object. When the interface program is processed, an alternative identifier is determined by reference to the class identifier, the alternative identifier specifying an alternative class to which the application object in the network element is assigned. The alternative identifier is incorporated into an amended  
10 maintenance message. When the amended maintenance message is processed by an application program, the application object is then processed as belonging to the alternative class. This is possible because the application object is allomorphic with respect to the class which is known in the service computer and which is required for the application object in the unamended maintenance message by the  
15 service computer.

The interface program performs the assignment of the alternative identifiers to the class identifiers centrally for a number of application programs. In the method according to the present invention, this step does not have to be programmed in each application program, but rather only once in the interface  
20 program. Given several hundred application programs per control computer, this reduces the complexity of programming, maintenance and documentation considerably. The application programs are kept free of additional steps which are necessary when taking allomorphy into consideration because these steps are carried out centrally in the interface program which is situated upstream.

25 Some of the additional steps are also carried out in databases which are situated downstream and which are utilized by the application programs.

The execution of the assignment of alternative identifiers and auxiliary identifiers in a central interface program is possible because allomorphy is defined at the class level in the method according to the present invention. Such a  
30 definition is not mentioned in the standard X.720, but is nevertheless compatible with the standard. Allomorphy at the class level refers to means that all the objects

of the alternative class being carried out as if they were objects of the original class. A definition of allomorphy which is referred to all the objects of the alternative class then does not result in any disadvantages if predefined programming rules are complied with. Examples of such programming rules are explained below in  
5 relation to the exemplary embodiments.

The method according to the present invention makes it possible to comply easily with the stipulations of the standard X.720. The application programs in the control computer can be developed with a small amount of additional expenditure, in which case it is always ensured that no errors occur when the control network is  
10 operated even when programs remain unchanged in the service computer.

In one embodiment, a table with which the alternative identifiers are assigned to the class identifier is used in the interface program. The table is stored in the memory of the control computer. An entry in the table is read in that a memory cell which is assigned to the class identifier and contains the alternative  
15 identifier belonging to the class identifier is read. To determine the alternative identifier, all that is therefore required is a single read access to the memory. If the alternative identifiers change as a result of developments of the application programs, all that is necessary is to reprogram the memory contents. As such, the contents of the table can be easily replaced or expanded.

If, in another embodiment, after the amended maintenance message has been processed, the application program generates a confirmation message in which the class specified during the generation of the application object to be processed is specified as a class identifier, the confirmation message subsequently can be processed by the interface program. For example, when the interface program is  
25 processed, it is possible to determine by reference to the class identifier which data are to be removed from the confirmation message. To do this, the table used in the interface program is expanded to such an extent that entries to the permitted data are also associated with each class identifier. The interface program then generates from the confirmation message a new confirmation message which contains only  
30 such data of an application object of the class to which the confirmation message refers.

The class specified when the application object to be processed is generated is stored in one embodiment as an origin class in the data of the application object to be processed. When the application program is processed, the origin class is then used as a class identifier. This procedure makes the origin class available easily.

5 If, in another embodiment, the confirmation message also contains an auxiliary identifier in which at least one class is designated which is known in the service computer and/or in at least one other service computer as the class to which the application object to be processed belongs, the program in the service computer can determine later, by reference to the auxiliary identifier, how the received  
10 confirmation message is to be processed. This is significant, in particular, if the class specified in the class identifier of the confirmation message received by the service computer is not yet known in the service computer. The service computer then determines the class to which the confirmation message refers by reference to the class or classes specified in the auxiliary identifier. The auxiliary identifier  
15 contains, in other words, the classes with respect to which the application object is allomorphic. If the confirmation message contains not only the class identifier but also an origin identifier in which the origin class is specified, the requirements of the protocol for the exchange of messages in the control computer and also for the protocol for the exchange of messages between the service computer and  
20 control computer can be fulfilled.

In another embodiment, at least one class which is known in the service computer and/or in at least one other service computer for the application object is stored as an allomorph class in the data of the application object. When the application program is processed, the allomorph class is then used as an auxiliary  
25 identifier. This measure results in an easy-to-manage data structure in which the application objects administer their allomorph classes themselves. It is not necessary to take any additional measures with respect to the allomorph class in the interface program or in the application program.

In another embodiment, the interface program is also responsible for other  
30 interface functions. For example, for the control of events for defining the processing sequence of the maintenance messages or for protocol adaptations of

these messages, referred to in English as "basic encoding". As a result of this measure, on the control computer there is just a single interface program which is programmed and maintained in a standardized fashion.

Additional features and advantages of the present invention are described in,  
5 and will be apparent from, the following detailed description of the invention and the figures.

### BRIEF DESCRIPTION OF THE FIGURES

Figure 1 shows part of a control network for controlling a telecommunications network.

10 Figure 2 shows the development of an original class A into an expanded class A', whose objects can be controlled as objects of the old class A during the operation of the control network.

Figure 3 shows the processing of messages in the control computer of a switching unit after the development, an object being controlled which has been  
15 generated before the development.

Figure 4 shows the processing of messages in the control computer after the development, an object being controlled which has been generated after the development.

Figure 5 shows the association of the names of classes A and A' and an  
20 access to objects of the two classes via a filter function.

### DETAILED DESCRIPTION OF THE INVENTION

Figure 1 shows part of a control network 10 for controlling a telecommunications network 12. The telecommunications network 12 contains a multiplicity of switching offices, of which the switching offices 14 and 16 are  
25 illustrated in Figure 1. The telecommunications network 12 also includes connecting lines between the switching offices, of which a connecting line 18 between the switching office 14 and the switching office 16 is illustrated in Figure 1. The telecommunications network 12 connects the subscribers of the telecommunications network 12; for example, a subscriber Tln1 connected to the  
30 switching office 14 and a subscriber Tln2 connected to the switching office 16.

The control network 10 contains dedicated transmission links 20 and 22 which lead to a service computer 24. The transmission link 20 transmits maintenance messages from the service computer 24 to the switching office 14 in order, for example, to change subscriber data of the subscriber Tln1 in the  
5 switching office 14. The switching office 14 itself transmits confirmation messages to the service computer 24 in order to signal satisfactory processing of the received maintenance message. The transmission link 22 serves for bidirectional transmission of data between the service computer 24 and switching office 16.

The maintenance messages are processed in the switching office 14 by a  
10 control computer 34 and in the switching office 16 by a control computer 36. The data structures to which the maintenance messages refer belong to the same class A in the service computer 24 and in the switching office 14. The switching office 16 on the other hand contains data structures of a class A', which have been developed in comparison with the class A required in the service computer 24. Error-free  
15 operation of the control network 10 is ensured with respect to the switching office 16 by virtue of the fact that allomorphy has been taken into consideration in the development of the class A into class A'. What allomorphy refers to in this context is explained below with reference to Figure 2.

In the exemplary embodiment illustrated in Figure 1, the class A which  
20 defines, for example, the data structure of the subscriber data, for example the call number and the useable services of the telecommunications network 12, is used in the switching office 14. The subscriber data of the subscriber Tln1 is stored in an object a1 in accordance with the data structure predefined by the class A in a memory 38 of the control computer 34. The class A is also known in the service  
25 computer 24, indicated by the letter A in a memory 40 of the service computer 24.

In the switching office 16, the class A was developed into the class A'. An object a2 contains, for example, the subscriber data of the subscriber Tln2. The object a2 was firstly stored in the memory 42 before the class A was developed into the class A'. However, in the development, the original object a2 was converted,  
30 specifically into an expanded object a2 of the class A' by supplementing a data field. An object a3 in the memory 42 belongs to the class A' and contains the

subscriber data of a subscriber Tln3 whose connection was not set up in the switching office 16 until after the development. Although the programs in the service computer 24 only support objects of the class A, the objects a2 and a3 belonging to the class A' can be interrogated, amended or newly set up as objects of the class A from the service computer 24. The expansions of the class A', in comparison to the class A cannot, however, be processed by the service computer 24 until the programs in the service computer 24 have been amended at a later time in such a way that the class A' is also known in the service computer 24.

Figure 2 shows the classes A and A' and the original object a2 and the object a3. Below, the meaning of the designation "allomorphous with respect to" is explained in conjunction with Figure 2. The class A' differs from the class A only by having an additional data field 50. The data structure of the class A has therefore been expanded by the data field 50 in order to be able to take into consideration a further property of the subscribers Tln2, Tln3 during the operation of the switching office 16; for example, whether the subscriber Tln2, Tln3 is connected to the switching office 16 via an optical waveguide or via a copper conductor. For this reason, the class A' is also designated below as an expanded class A'. A data field 50' in the object a3 contains, as data item, a value which indicates that the subscriber Tln3 is connected to the switching office 16 via an optical waveguide. The object a3 is generally designated as an expanded object a3.

The class which is specified when an object is generated is designated as an original class of this object. The object a2 had, as original class, the class A indicated by an arrow 52. On the other hand, the expanded object a3 has, as original class, the expanded class A' indicated by an arrow 54.

A first possible way of defining the data structure of the class A includes generating the class A' from the class A via what is referred to as inheritance which is defined in object-oriented programming languages. Such programming languages are, for example, the languages C++ and CHILL. When inheritance occurs, the programmer specifies that the expanded class A' is to take over all the data structures and what are referred to as methods for processing the data structures from the class A. It is also specified that the class A' additionally

contains the data field 50. Another possible way of defining the class A' is to  
redefine this class. In this case, the class A' is defined in the way in which it was  
already defined in class A. However, in addition, the data field 50 is also defined.  
The relationship between the corresponding parts of class A and of the expanded  
5 class A' is illustrated in Figure 2 via dashed lines 56.

An imaginary object a3\* contains, from the object a3, all the data fields and  
all the methods for processing the data fields which also would have been generated  
when the subscriber Tln3 is set up before the development, when the class A  
existed, but the class A' did not yet exist. In the object a3\*, there is for this reason  
10 no data field corresponding to the data field 50 or 50'. This fact is indicated by  
dashed lines 58. The object a3\* is a visual aid for delimiting the terms "compatible  
with" and "allomorphic with respect to". An arrow 60 indicates that the object  
a3\* is compatible with the class A because it has precisely the data structures which  
are predefined in the class A. The expanded object a3 is on the other hand  
15 allomorphic with respect to the class A, cf. arrow 62. The object a3 has the  
allomorphic class A.

Allomorphy is the ability of the objects in class A' to be controlled as if they  
were objects of their allomorphic class A if this ability arises as a result of  
measures on the part of the application program. In an incremental expansion it is  
20 possible for there to be more than one allomorphic class; for example, the  
allomorphic class of the last expansion and the allomorphic class of the  
penultimate expansion.

An expanded object has only one allomorphic class if the expanded object  
is compatible in accordance with standard X.720 section 5.2.2 without the  
25 expansions to form the allomorphic class. In particular, the expanded object  
accordingly has all the attributes, attribute groups, control functions and  
confirmation methods which are also defined in the allomorphic class. Making  
allowance for allomorphy in the expansion of the class A ensures that the control  
network 10 also operates without errors after the expansion.

30 Figure 3 shows the processing of messages in the control computer 16 after  
the development of the class A into the class A', the object a2 which has been



configured before the development as belonging to the class A being controlled. During the development of the class A into the class A', allomorphy was taken into account so that objects of the class A' are allomorphous with respect to the class A. In addition, in the development all the objects of the class A were converted into  
5 objects of the class A' through supplementing data fields and methods.

The control computer 16 contains an interface program 100 which processes maintenance messages coming from the service computer 24, for example a maintenance message WN1, and which transmits confirmation messages, for example a confirmation message BN1', to the service computer 24. The interface  
10 program 100 is the interface between the service computer 24 and a number of application programs in the control computer 16, of which two application programs 102 and 104 are shown in Figure 3. The application program 102 serves to administer the data associated with the subscribers Tln2, Tln3 connected to the switching office 16. The application program 104 is used for traffic measurement.

15 The maintenance message WN1 coming from the service computer is passed on, as amended maintenance message WN1', to the application program 102 when the interface program 100 is processed. On the other hand, maintenance messages which are intended for the application program 104 are passed on to the application program 104, cf. arrow 106, when the interface program 100 is  
20 processed.

After the processing of the maintenance message WN1' in the application program 102, the application program 102 generates a confirmation message BN1 for the interface 100. If the application program 104 has processed a maintenance message, it also transmits a confirmation message to the interface program 100, cf.  
25 arrow 108.

During the processing of the maintenance message WN1', the application program 102 cooperates with a database program 110 which is also present in the control computer 36 and is used to store, change, delete or read subscriber data in the memory 42. The application program 102 transmits requests in the form of  
30 messages to the database program 110; for example, the message N1. After the request in the message N1 has been executed, the database program 110 transmits a

result message EN1 back to the application program 102. The application program 104 cooperates with its own database program 112 by transmitting requests to the database program 112, cf. arrow 114, and by receiving and further processing result messages from the database program 112, cf. arrow 116.

5           The database program 110 uses the same access method for the objects of the original class A and the objects of the developed class A'. This is possible because combination classes are used in which the data structures and methods of the original class and of the expanded class are combined. A combination class KA is the combination of the class A and of the class A'. All the objects in the memory 10 42 which have the allomorphic class A as original class are expanded during the development by the additional data fields of the expanded class A. The additional data fields are filled with predefined values.

          The maintenance message WN1 contains a class identifier moC which specifies the class A as the class to which the maintenance message WN1 refers.  
15   An object identifier moI specifies the object a2 to which the maintenance message WN1 refers. The maintenance message WN1 is transmitted by the service computer 24 in order to find out the subscriber data of the subscriber Tln2. All that is known in the service computer 24 is that this subscriber data is contained in the object a2 which is stored in the control computer 36. The maintenance message 20 WN1 contains further data fields which are not illustrated. The read operation to be carried out, for example, is defined in one of these data fields.

          When the maintenance message WN1 is processed in the interface program 100, the class A specified in the class identifier moC is determined. When the interface program 100 is processed, the class A' is determined as an alternative 25 class via a table T by reference to the class A which is determined in this way, and the class A' is entered into the class identifier moC of the maintenance message WN1. The table T is stored in a memory 122 of the control computer 36.

          The maintenance message WN1' relates, with the class identifier moC=A', to the expanded class A'. The value of the object identifier moI=a2 remains 30 unchanged in the maintenance message WN1'. An identifier allo in the maintenance message WN1' specifies all the classes which are allomorphic with

respect to the class A', i.e. in the exemplary embodiment the class A. The interface program 100 also obtains these classes from the table T. The other data of the maintenance message WN1 are transferred into the maintenance message WN1'.

When the maintenance message WN1' belonging to the maintenance message WN1  
5 is generated, the interface program 100 also carries out protocol adaptation of a transmission protocol on the transmission link 22 into a message protocol which is used within the control computer 36.

The message N1 which is generated by the application program 102 during the processing of the maintenance message WN1' contains an instruction B1 which  
10 specifies that data is to be read. As a parameter of the instruction B1, the message N1 contains the class A' to which the data to be read belongs, as well as the object a2 whose data is to be read. The application program 102 processes exclusively messages which refer to objects of the class A'. No further measures are taken in the application program 102 with respect to the class A.

15 The database program 110 accesses the memory 42 during the processing of the message N1 in order to read the data of the subscriber Tln2 which is stored in the object a2. The object a2 also contains an origin identifier oC in which the class which has been specified when the object a2 was generated is specified. The database program 110 reads the data requested with the instruction B1 in the object  
20 a2 and enters this data into the result message EN1. In addition, in the result message EN1 there is a note, via a response identifier AB1, that the result message EN1 contains the result which has been generated when the instruction B1 was processed.

Furthermore, in the result message EN1 the class A' is specified as the class  
25 which has been affected by the result message EN1. The origin identifier oC=A is also transferred from the database program 110 in the result message EN1 to the application program 102.

When the application program 102 is processed it is a prerequisite that all the messages to be processed refer to the class A' and not to the class A. The value  
30 of the origin identifier oC=A is transferred as a value of the class identifier moC in the confirmation message BN1 by the application program 102. For this value

assignment it is not necessary for the application program 102 to be able to process objects of the class A. In addition, the confirmation message BN1 contains the origin identifier oC=A and the interrogated subscriber data of the subscriber Tln2. The interface program 100 does not need to remember the class identifier of the maintenance message WN1 by virtue of this procedure.

The confirmation message BN1 is processed by the interface program 100 and transmitted, as confirmation message BN1', to the service computer 24 on the transmission link 22 in accordance with the transmission protocol. By reference to the table T, the interface program 100 determines which data fields in the message BN1 are not contained in objects of the class A. These data fields are not transferred into the confirmation message BN1'.

The service computer 24 receives the confirmation message BN1' and can process it as a message which refers to an object of the class A. Objects of the class A' in the control computer 36 are controlled from the service computer 24 as if they were objects of the class A. The development in the control computer 36 does not, therefore, restrict the operating features of the service computer 24.

Figure 4 shows the processing of messages in the control computer 16 after the development of the class A into the class A', the object a3 which is not generated until after the development being controlled.

Via inputs by an operator into the service computer 24, a maintenance message WN2 is generated in the service computer 24, the maintenance message WN2 having the function of generating in the control computer 16 the object a3 for the subscriber data of the subscriber Tln3. For this reason, the maintenance message WN2 contains a coding for the instruction "Generate" in an instruction field. The class identifier moC in the maintenance message WN2 identifies the class A as the class to which the maintenance message WN2 refers. The object identifier moI of the maintenance message WN2 identifies the object a3 which is to be generated via the maintenance message WN2. The maintenance message WN2 also contains further data which is not illustrated.

The interface program 100 processes the maintenance message WN2 in the same way as explained above for the maintenance message WN1. When the

maintenance message WN2 is processed, the table T stored in the memory 122 is again used to determine the alternative class A' for the class A specified in the class identifier moC of the maintenance message WN2 and to use it as the class identifier moC of the maintenance message WN2'. The object identifier moI=a3 has the  
5 same value in the maintenance message WN2' as in the maintenance message WN2. The remaining data is also transferred from the maintenance message WN2 into the maintenance message WN2' when the interface program 100 is executed. In addition, in the identifier allo of the maintenance message WN2', it is noted that the class A is the class with respect to which the class A' is allomorphic. The  
10 maintenance message WN2' generated in this way is transmitted from the interface program 100 to the application program 102 in accordance with the protocol in the control computer 36.

When the maintenance message WN2' is processed, the application program 102 transmits a message N2 to the database program 110 in order to allow the  
15 object a3 to be created in the memory 42. The message N2 contains, in encrypted form, the instruction "Generate", the expanded class A' and the name of the object a3 to be created.

When the message N2 is processed, the object a3 is generated in the memory 42 via the database program 110 by assigning to this object specific  
20 memory cells which are filled with initial values. In the origin identifier oC of the object a3 there is a note of the class A' because this class was specified when the object a3' was generated. An identifier allo in the object a3 refers to the class A because the object a3 is allomorphic with respect to the class A.

The database program 110 generates a result message EN2 in order to  
25 confirm the generation of the object a3. The result message EN2 contains a response identifier AB2 which specifies that the result message EN2 was produced when an instruction "Generate" was produced. In addition, the result message EN2 contains an identifier which refers to the class A', the origin identifier oC=A', the identifier allo=A and further data (not illustrated) of the object a3.

30 The result message EN2 is processed by the application program 102, a confirmation message BN2 being generated. The value of the origin identifier

oC=A' is used as a value of the class identifier moC in the confirmation message BN2. The other data items in the result message EN2 are transferred into the confirmation message BN2.

When the interface program 100 is processed, after the reception of the  
5 confirmation message BN2, a confirmation message BN2' is generated in  
accordance with the transmission protocol used on the transmission link 22. The  
confirmation message BN2' contains all the data of the confirmation message BN2  
because the interface program 100 determines, by reference to the table T, that it is  
not necessary to remove any data fields if the confirmation message BN2 has the  
10 class A' as class identifier.

During the processing of the confirmation message BN2' in the service  
computer 24, it is determined by reference to the identifier allo that the  
confirmation message BN2' relates to data of an object of the class A or that the  
confirmation message BN2' can be treated as if it contained data of an object of the  
15 class A. Although the program in the service computer 24 has not been changed  
after development in the control computer 36, objects of the class A' in the control  
computer 36 can be controlled from the control computer 24 as if they were objects  
of the class A. As long as only the class A is known in the service computer A, it is  
also the case that only the data items contained in objects of the class A are  
20 processed in the confirmation message BN2'.

Via the method explained with reference to Figures 3 and 4, it is also  
possible to support classes which are allomorphous with respect to a number of  
classes. If, for example, the class A' is developed into a class A'', objects can be  
allomorphous with respect to the classes A' and A. The control computer 36 then  
25 can be executed by service computers 24 during whose programming at least one of  
the classes A, A' or A'' was known. It is possible to dispense with the class A as  
soon as all the service computers know at least the class A'.

The method explained with reference to Figures 3 and 4 is applied in  
maintenance messages for generating objects, and in maintenance messages for  
30 reading objects. In addition, this method can be used to change data in objects and

delete objects. For these methods to operate without errors, the service computer 24 must be capable:

- of assigning the confirmation messages BN to the associated maintenance messages WN,
- 5       - of reading and evaluating the identifier allo in the confirmation messages BN,
- of skipping over unknown name associations (cf. explanations relating to Fig. 5) and unknown identifier values and parameters,
- of storing the origin class of an object if there is a database in the service computer,
- 10       - of skipping over unknown optional values, and
- of skipping over unknown values of the "enumerated" data type.

In other words, the service computer 24 must be programmed in such a way that objects can be transferred from it into a control computer with a greater degree of knowledge.

15

The interface program 100 has the following features:

- When a class is expanded it is necessary to incorporate the expanded class into the table T, such that for each expanded class it is necessary to store the classes with respect to which the expanded class is allomorphous.
- 20       - The name associations which have been made by the expanded class must be stored in the table T.
- The interface program 100 is to remove from the confirmation messages BN coming from the application program 102 the data which does not belong to the compatible class if the confirmation message BN transmitted to the service computer 24 does not relate to the non-expanded class.
- 25       - The interface program 100 must be able to process filter instructions.

This is explained in more detail below with reference to Figure 5.

The application program 102 fulfills the following requirements:

- As soon as the system moves over to the expanded class, only objects of the new class are then generated, irrespective of the knowledge of the service computer.
- 5        - The new classes are either expanded classes or classes which have nothing to do with the previous classes.

The database program 110 is configured in such a way that, after a development, the entire data stock relating to the origin class is converted into a data stock of the expanded class.

- 10        The creation of the table T and the conversion of the data stock in the database of the database program 110 are supported by service programs. These service programs evaluate description languages which are used to specify the expansion of classes.

- 15        Conditions which have to be complied with when taking into consideration allomorphy are dealt with below. These conditions apply on the level of the classes although, according to standard X.720, allomorphy is firstly a property of an object. In the standard X.720, the following conditions are mentioned:

- conditions for the expanded class in section 5.2.2.1,
- conditions for what are referred to as program packages in the sections 5.2.2.1 and 5.2.2.2,
- 20        - conditions for identifier values in the sections 5.2.2.3 and 5.3.4.1,
- conditions for what are referred to as identifier groups in section 5.2.2.4,
- conditions for actions, confirmations and parameters in section 5.2.2.54,
- 25        - conditions for the behavior of the objects in section 5.2.2.6, and
- conditions for the name association in section 5.3.4.1.

Figure 5 shows, in a part a, a section of what is referred to as a binary tree which defines the name association of objects of the class A and A'. Name



association is the assignment of the object to what is referred to as a superordinate object, the assignment being used for the definition of a uniquely defined name for an object. Identical object names for different objects can be used if the objects each belong to another superordinate object. A name which is uniquely defined in the control computer 16 is then produced from the name of the superordinate object and the name of the objects which are subordinated in this way. The superordinate object is to be specified when an object is generated. The subordinate object is considered as being contained in the superordinate object, referred to in English as "containment".

10       An object b1 of the class B is the superordinate object for the objects a2 and a3, which both belong to the class A' after the development, but have different original classes  $oC=A$  and  $oC=A'$ . The name associations are noted in the table T which is used by the interface program 100.

15       For the portion of the name tree shown in part A of Figure 5 there is a note in table T that objects of the class B contain subordinate objects of the original class A and of the class A'. At each development, the name associations in the table T are also adapted to the name tree which is also decisive after the development.

20       In part b of Figure 5, an access to objects of both classes A and A' is illustrated after the service computer 24 also knows both classes A and A'. A maintenance message WN3 contains a filter instruction  $F1=((oC=A) \text{ OR } (oC=A'))$  in which it is defined that subordinate objects of the class A or A' are to be registered. The class identifier  $moC=B$  of the maintenance message WN3 specifies that the maintenance message WN3 also refers to the class B. The object identifier  $moI=b1$  of the maintenance message WN3 specifies that the object b1 is to be processed; i.e., is the superordinate object.

25       When the interface program 100 is executed in the switching office 16, the filter instruction F1 is transmitted unchanged to the application program 102. As such in particular, that operations which refer to the original class A do not have to be replaced in the filter instruction F1 by operations which refer to the expanded class A'. This measure ensures that the service computer 24 can distinguish between objects of the original class A and of the original class A'.

5 The application program 102 causes the data of the objects a2 and a3 whose original class is the class A and A', respectively, to be read from the memory 42 via database program 110. Two confirmation messages BN3 and BN4 (not illustrated) are transmitted to the interface program 100. A new confirmation message BN3', which contains the class identifier moC=A, the object identifier moI=a2, the origin identifier oC=A and the auxiliary identifier allo={} is generated from the confirmation message BN3 in the interface program 100. A confirmation message BN4', which contains the class identifier moC=A', the object identifier moI=a3, the origin identifier oC=A' and the auxiliary identifier allo={A} is generated from the confirmation message BN4 in the interface program 100.

15 The CCITT standard X.734 "Information Technology - Open Systems Interconnection - Systems Management: Event Report Management Function" from 1993 explains event control in the control network 10, cf. Figure 1. What are referred to as discriminators are used which pass on events to the service computer 24 within the switching office 16 only under certain conditions. After the development of the class A into the class A', it is sufficient to convert the discriminators which relate to the class A into discriminators which relate to the class A'. If new discriminators are generated after the development, the class A is replaced by the class A' if the class A is specified as a selection criterion for the passing on of the messages.

20 Although the present invention has been described with reference to specific embodiments, those of skill in the art will recognize that changes may be made thereto without departing from the spirit and scope of the invention as set forth in the hereafter appended claims.

## 25 ABSTRACT OF THE DISCLOSURE

A method for operating a telecommunications network wherein a network element at a network node of the telecommunications network is controlled by a control computer, the control computer is maintained from a service computer, allomorphy is taken into account in the development of application programs so that even a service computer which is not developed can maintain the control computer, and making allowance for allomorphy results in little additional

expenditure because an interface program is used in which essential processing steps are carried out for all the application programs, the processing steps ensuring that allomorphy is taken into consideration.

In the Claims:

- 5           On page 24, cancel line 1, and substitute the following left-hand justified heading therefor:

CLAIMS

          Please cancel claims 1-17, without prejudice, and substitute the following claims therefore:

- 10           18.    A method for operating a telecommunications network, the method comprising the steps of;
- controlling a network element at a network node of the telecommunications network via a control computer;
- storing at least one interface program and a plurality of application
- 15    programs in the control computer in addition to an operating system;
- processing application objects during execution of the application programs, the application projects having, depending on their membership of a class, data with a predefined data structure and predefined methods for processing the data;
- setting up a link between a service computer and a control computer,
- 20    maintenance being performed on the control computer via the link by at least one maintenance message;
- processing the maintenance messages, via the interface program, coming from the service computer, each of the maintenance messages containing a class identifier which assigns the maintenance message to a class, the class identifier
- 25    specifying the class, known in the service computer, of an application object to be processed;
- determining an alternative identifier, when the interface program is processed with reference to the class identifier, which indicates an alternative class to which the application object to be processed belongs in the network element, the
- 30    alternative identifier being incorporated into an amended maintenance message;
- and

processing the amended maintenance message, wherein the application object to be processed is processed as an object of the alternative class by an application program.

5           19.     A method for operating a telecommunications network as claimed in Claim 18, wherein when the alternative identifier is determined, a first table stored in a memory of the control computer is used, in which table the alternative identifier is assigned to the class identifier.

10           20.     A method for operating a telecommunications network as claimed in Claim 18, the method further comprising the step of:

generating a confirmation message, via the application program and after the processing of the amended maintenance message, in which the class specified when the application object to be processed to be generated is specified.

15           21.     A method for operating a telecommunications network as claimed in Claim 20, the method further comprising the steps of:

generating an amended confirmation message from the confirmation message, when the interface program is processed, the amended confirmation  
20     message containing only data which has an application object of the class to which the confirmation message relays; and  
transmitting the amended confirmation message to the service computer.

22.     A method for operating a telecommunications network as claimed in  
25     Claim 20, the method further comprising the steps of:

storing the class specified, when the application object to be processed is generated, as an origin class in the data of the application object to be processed; and

using the origin class as a class identifier when the application program is  
30     processed.

23. A method for operating a telecommunications network as claimed in Claim 20 wherein the confirmation message contains an identifier in which at least one class is designated which is known in the service computer as the class to which the application object to be processed belongs.

5

24. A method for operating a telecommunications network as claimed in Claim 23, wherein the confirmation message contains, in addition to the class identifier, an origin identifier in which the origin class is specified.

25. A method for operating a telecommunications network as claimed in Claim 23, the method further comprising the steps of:

storing at least one class known in at least one service computer for the application object as an allomorph class in the data of the application object; and using the allomorph class as an auxiliary identifier when the application program is processed.

15

26. A method for operating a telecommunications network as claimed in Claim 21, wherein a confirmation message which is generated for the service computer when the interface program is being processed contains the class identifier and the origin identifier, and possibly an identifier in which at least one class is designated.

20

27. A method for operating a telecommunications network as claimed in Claim 18, wherein the network element is selected from the group consisting of a switching office, a cross-connector and a concentrator unit.

25

28. A method for operating a telecommunications network as claimed in Claim 18, wherein the telecommunications network is selected from the group consisting of a fixed network, a mobile radio network, and a network with a fixed network component and a mobile radio network component.

30

29. A method for operating a telecommunications network as claimed in Claim 18, wherein the interface program carries out further interface functions between the service computer and the application programs, including at least one of an events control for defining a processing sequence of the maintenance  
5 messages, and adaptation of the messages coming from the service computer to a protocol for transmitting messages within the control computer, and an adaptation of the confirmation messages coming from the application programs to a predefined protocol for transmitting messages between the service computer and the control computer.

10

30. A method for operating a telecommunications network as claimed in Claim 18, the method further comprising at least one of the following steps:

using a first application program for subscriber administration;

using a second application program for administering connecting lines to

15 other switching devices;

using a third application program for performing maintenance on the switching devices; and

using a fourth application program for measuring traffic on the switched links.

20

31. A method for operating a telecommunications network as claimed in Claim 30, wherein the application objects of the first application program contain the subscriber data for one subscriber in each case.

25 32. A method for operating a telecommunications network as claimed in Claim 18, wherein the maintenance messages contain a name identifier for a name of the application object to which the maintenance message relays.

30 33. A network element for operating a telecommunications network, comprising:

a control computer for controlling the network element, at least one interface program and a plurality of application programs being stored in the control computer in addition to an operating system, application objects being processed during execution of the application programs, the application objects having, depending on their membership of a class, data with a predefined data structure and predefined methods for processing the data; and

a service computer connected to the control computer via a link, maintenance being performed on the control computer by at least one maintenance message, the interface program processing the maintenance message is coming from the service computer, each of the maintenance messages containing a class identifier which assigns the maintenance message to a class, the class identifier of the maintenance message specifying the class, known in the service computer, of an application object to be processed, when the interface program is processed with reference to the class identifier an alternative identifier is determined which indicates an alternative class to which the application object to be processed belongs in the network element, the alternative identifier being incorporated into an amended maintenance message, and when the amended maintenance message is processed, the application object to be processed is processed as an object of the alternative class by an application program.

20

34. A telecommunications network comprising a plurality of network elements, wherein at least one of the plurality of network elements is for operating the telecommunications network, the network elements comprising:

a control computer for controlling the network element, at least one interface program and a plurality of application programs being stored in the control computer in addition to an operating system, application objects being processed during execution of the application programs, the application objects having, depending on their membership of a class, data with a predefined data structure and predefined methods for processing the data; and

a service computer connected to the control computer via a link, maintenance being performed on the control computer by at least one maintenance

message, the interface program processing the maintenance message is coming from the service computer, each of the maintenance messages containing a class identifier which assigns the maintenance message to a class, the class identifier of the maintenance message specifying the class, known in the service computer, of an application object to be processed, when the interface program is processed with reference to the class identifier an alternative identifier is determined which indicates an alternative class to which the application object to be processed belongs in the network element, the alternative identifier being incorporated into an amended maintenance message, and when the amended maintenance message is processed, the application object to be processed is processed as an object of the alternative class by an application program.

#### REMARKS

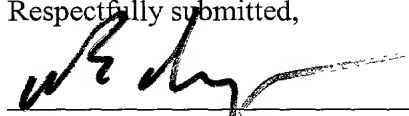
The present amendment makes editorial changes and corrects typographical errors in the specification, which includes the Abstract, in order to conform the specification to the requirements of United States Patent Practice. No new matter is added thereby. Attached hereto is a marked-up version of the changes made to the specification by the present amendment. The attached page is captioned "Version With Markings To Show Changes Made".

In addition, the present amendment cancels original claims 1-17 in favor of new claims 18-34. Claims 18-34 have been presented solely because the revisions by red-lining and underlining which would have been necessary in claims 1-16 in order to present those claims in accordance with preferred United States Patent Practice would have been too extensive, and thus would have been too burdensome. The present amendment is intended for clarification purposes only and not for substantial reasons related to patentability pursuant to 35 U.S.C. §§103, 102, 103 or 112. Indeed, the cancellation of claims 1-17 does not constitute an intent on the part of the Applicants to surrender any of the subject matter of claims 1-17.



Early consideration on the merits is respectfully requested.

Respectfully submitted,



(Reg. No. 39,056)

5

William E. Vaughan  
Bell, Boyd & Lloyd LLC  
P.O. Box 1135  
Chicago, Illinois 60690-1135  
(312) 807-4292  
Attorneys for Applicants

10

CONFIDENTIAL

**VERSIONS WITH MARKINGS TO SHOW CHANGES MADE  
In The Specification:**

The Specification of the present application, including the Abstract, has been amended as follows:

5 SPECIFICATION

~~Method and network element for operating a telecommunications network~~

TITLE OF THE INVENTION

METHOD AND NETWORK ELEMENT FOR  
OPERATING A TELECOMMUNICATIONS NETWORK

10 BACKGROUND OF THE INVENTION

Description

20 The present invention relates to a method for operating a telecommunications network, in which a network element at a network node of the telecommunications network is controlled by a control computer. The network element is, for example, a switching office for switching links, what is referred to as a cross-connector or a concentrator unit for connecting a plurality number of subscribers to a connecting line. In addition to the operating system for operating the control computer, a plurality number of application programs, during whose execution application objects are processed, are stored in the control computer. The application objects are associated with data with a predefined data structure and preferably also predefined methods for processing the data. The data structure and the methods are dependent on a class which is also to be specified when the respective application object is generated. A link via which the control computer is maintained by means of via maintenance messages is set up between a service computer and a control computer.

25 Such methods are used for controlling the telecommunications network if, for example, a new switching device is to be put into operation as a network element or if subscriber data in the switching device have to be changed at a later date, such as is the case when new subscribers are connected or when an existing subscriber moves house. Efficient methods for controlling the telecommunications network are obtained if what are referred to as open systems are used which are programmed in compliance with standards which apply worldwide. For example,

standards of the ISO (International Standardization Organization) and of the ITU (International Telecommunication Union) with its body the ITU-T, known earlier as the CCITT (International Telegraph and Telephone Consultative Committee) relate to the setting up of such open systems. A separate control network is to be used to control the telecommunications network. The interfaces between the service computer and switching device are standardized in protocols Q1, Q2 and Q3.

The application objects are defined as objects of an object-oriented language, for example in the language C++ or CHILL. If the application programs are developed, it is necessary to ensure that the control network also operates without faults with the new application programs. ~~This means~~ As such, in particular ~~that~~, application objects which are considered by the service computer as belonging to an original class cannot readily be assigned to an amended alternative class.

This problem is mentioned in the CCITT standard X.720 (01/92) -

“Information Technology - Open Systems Interconnection - Structure of Management Information: Management Information Model” - in the section 5.2.1. In the section 5.2.3 of the standard X.720, two methods for solving the problem are proposed. In the first method, on the part of the application program, a programming technique is used which takes allomorphy into consideration.

Allomorphy is the ability of a specific application object of the alternative class to be controlled as if it were an object of the original class if this ability arises as a result of measures on the part of the application program. The other method ~~comprises~~ includes taking measures on the part of the service computer which permit cooperation between the service computer and the application program even when the application program is developed.

European patent application EP-A-0 817 422 discloses a method for implementing controlled objects in a subsystem of a controlled system in a network, at least one control system and one controlled system being present. The controlled objects are implemented independently of other subsystems without knowing the type of the objects in the other subsystems. They can be connected to other objects and can transmit messages to them. For this purpose, a first object is

generated for cooperating with an abstract object. The abstract object has a defined interface which is called up using the first object and which inherits the abstract object from a second object, the generated second object being unknown to the first, and the second object being intended for cooperation with the first object. The first object which cooperates with the second object considers the second object as an object of the abstract type. However, in this method known from EP-A-0 817 422, there is no use of programming technology which takes into consideration the allomorphy in accordance with the section 5.2.3 of the standard X.720.

The An object of the present invention, therefore, is to diselose provide a simple method for operating a telecommunications network in which allomorphy is taken into consideration.

~~This object is achieved by means of a method having the features of patent claim 1. Developments are disclosed in the subelaims.~~

#### SUMMARY OF THE INVENTION

In the method according to the present invention, in each case a class identifier in which the class to which the maintenance message relates is specified is determined from the maintenance messages during the processing of an interface program which is used for a plurality number of application programs. The class identifier in the maintenance message specifies the class, known in the service computer, of an application object to be processed. As a result of the development, the class known in the service computer sometimes deviates from the actual class of the application object. When the interface program is processed, an alternative identifier is determined by reference to the class identifier, ~~said~~ the alternative identifier specifying an alternative class to which the application object in the network element is assigned. The alternative identifier is incorporated into an amended maintenance message. When the amended maintenance message is processed by an application program, the application object is then processed as belonging to the alternative class. This is possible because the application object is allomorphous with respect to the class which is known in the service computer and

which is required for the application object in the unamended maintenance message by the service computer.

The interface program performs the assignment of the alternative identifiers to the class identifiers centrally for a plurality number of application programs. In  
5 the method according to the present invention, this step does not have to be programmed in each application program, but rather only once in the interface program. Given several hundred application programs per control computer, this reduces the complexity of programming, maintenance and documentation considerably. The application programs are kept free of additional steps which are  
10 necessary when taking allomorphy into consideration because these steps are carried out centrally in the interface program which is situated upstream.

Some of the additional steps are also carried out in databases which are situated downstream and which are utilized by the application programs.

The execution of the assignment of alternative identifiers and auxiliary  
15 identifiers in a central interface program is possible because allomorphy is defined at the class level in the method according to the present invention. Such a definition is not mentioned in the standard X.720, but is nevertheless compatible with the standard. Allomorphy at the class level refers to means that all the objects of the alternative class ~~can be~~ being carried out as if they were objects of the  
20 original class. A definition of allomorphy which is referred to all the objects of the alternative class then does not result in any disadvantages if predefined programming rules are complied with. Examples of such programming rules are explained below in relation to the exemplary embodiments.

The method according to the present invention makes it possible to comply  
25 easily with the stipulations of the standard X.720. The application programs in the control computer can be developed with a small amount of additional expenditure, in which case it is always ensured that no errors occur when the control network is operated even when programs remain unchanged in the service computer.

In one ~~development~~ embodiment, a table with which the alternative  
30 identifiers are assigned to the class identifier is used in the interface program. The table is stored in the memory of the control computer. An entry in the table is read

in that a memory cell which is assigned to the class identifier and contains the alternative identifier belonging to the class identifier is read. To determine the alternative identifier, all that is therefore required is a single read access to the memory. If the alternative identifiers change as a result of developments of the application programs, all that is necessary is to reprogram the memory contents. This means that As such, the contents of the table can be easily replaced or expanded.

If, in another development embodiment after the amended maintenance message has been processed, the application program generates a confirmation message in which the class specified during the generation of the application object to be processed is specified as a class identifier, the confirmation message can subsequently easily can be processed by the interface program. For example, when the interface program is processed, it is possible to determine by reference to the class identifier which data are to be removed from the confirmation message. To do this, the table used in the interface program is expanded to such an extent that entries to the permitted data are also associated with each class identifier. The interface program then generates from the confirmation message a new confirmation message which contains only such data of an application object of the class to which the confirmation message refers.

The class specified when the application object to be processed is generated is stored in one development embodiment as an origin class in the data of the application object to be processed. When the application program is processed, the origin class is then used as a class identifier. This procedure makes the origin class available easily.

If, in another development embodiment, the confirmation message also contains an auxiliary identifier in which at least one class is designated which is known in the service computer and/or in at least one other service computer as the class to which the application object to be processed belongs, the program in the service computer can determine later, by reference to the auxiliary identifier, how the received confirmation message is to be processed. This is significant, in particular, if the class specified in the class identifier of the confirmation message

received by the service computer is not yet known in the service computer. The service computer then determines the class to which the confirmation message refers by reference to the class or classes specified in the auxiliary identifier. The auxiliary identifier contains, in other words, the classes with respect to which the application object is allomorphous. If the confirmation message contains not only the class identifier but also an origin identifier in which the origin class is specified, the requirements of the protocol for the exchange of messages in the control computer and also for the protocol for the exchange of messages between the service computer and control computer can be fulfilled.

In another ~~development~~ embodiment, at least one class which is known in the service computer and/or in at least one other service computer for the application object is stored as an allomorph class in the data of the application object. When the application program is processed, the allomorph class is then used as an auxiliary identifier. This measure results in an easy-to-manage data structure in which the application objects administer their allomorph classes themselves. It is not necessary to take any additional measures with respect to the allomorph class in the interface program or in the application program.

In another ~~development~~ embodiment, the interface program is also responsible for other interface functions. For example, for the control of events for defining the processing sequence of the maintenance messages or for protocol adaptations of these messages, referred to in English as "basic encoding". As a result of this measure, on the control computer there is just a single interface program which is programmed and maintained in a standardized fashion.

Additional features and advantages of the present invention are described in, and will be apparent from, the following detailed description of the invention and the figures.

~~The exemplary embodiments of the invention are explained below with reference to the drawings, in which:~~

#### BRIEF DESCRIPTION OF THE FIGURES

Figure 1 shows part of a control network for controlling a telecommunications network.

Figure 2 shows the development of an original class A into an expanded class A', whose objects can be controlled as objects of the old class A during the operation of the control network.

Figure 3 shows the processing of messages in the control computer of a switching unit after the development, an object being controlled which has been generated before the development.

Figure 4 shows the processing of messages in the control computer after the development, an object being controlled which has been generated after the development.

Figure 5 shows the association of the names of classes A and A', and an access to objects of the two classes ~~by means of~~ via a filter function.

#### DETAILED DESCRIPTION OF THE INVENTION

Figure 1 shows part of a control network 10 for controlling a telecommunications network 12. The telecommunications network 12 contains a multiplicity of switching offices, of which the switching offices 14 and 16 are illustrated in ~~figure~~ Figure 1. The telecommunications network 12 also includes connecting lines between the switching offices, of which a connecting line 18 between the switching office 14 and the switching office 16 is illustrated in ~~figure~~ Figure 1. The telecommunications network 12 connects the subscribers of the telecommunications ~~network 12,~~ network 12; for example, a subscriber Tln1 connected to the switching office 14 and a subscriber Tln2 connected to the switching office 16.

The control network 10 contains dedicated transmission links 20 and 22 which lead to a service computer 24. The transmission link 20 transmits maintenance messages from the service computer 24 to the switching office 14 in order, for example, to change subscriber data of the subscriber Tln1 in the switching office 14. The switching office 14 itself transmits confirmation messages to the service computer 24 in order to signal satisfactory processing of the received maintenance message. The transmission link 22 serves for bidirectional transmission of data between the service computer 24 and switching office 16.



The maintenance messages are processed in the switching office 14 by a control computer 34 and in the switching office 16 by a control computer 36. The data structures to which the maintenance messages refer belong to the same class A in the service computer 24 and in the switching office 14. The switching office 16  
5 on the other hand contains data structures of a class A', which have been developed in comparison with the class A required in the service computer 24. Error-free operation of the control network 10 is ensured with respect to the switching office 16 by virtue of the fact that allomorphy has been taken into consideration in the development of the class A into class A'. What allomorphy ~~means~~ refers to in this  
10 context is explained below with reference to ~~figure~~ Figure 2.

In the exemplary embodiment illustrated in ~~figure~~ Figure 1, the class A which defines, for example, the data structure of the subscriber data, for example the call number and the useable services of the telecommunications network 12, is used in the switching office 14. The subscriber data of the subscriber Tln1 is stored  
15 in an object a1 in accordance with the data structure predefined by the class A in a memory 38 of the control computer 34. The class A is also known in the service computer 24, indicated by the letter A in a memory 40 of the service computer 24.

In the switching office 16, the class A was developed into the class A'. An object a2 contains, for example, the subscriber data of the subscriber Tln2. The  
20 object a2 was firstly stored in the memory 42 before the class A was developed into the class A'. However, in the development, the original object a2 was converted, specifically into an expanded object a2 of the class A' by supplementing a data field. An object a3 in the memory 42 belongs to the class A' and contains the subscriber data of a subscriber Tln3 whose connection was not set up in the  
25 switching office 16 until after the development. Although the programs in the service computer 24 only support objects of the class A, the objects a2 and a3 belonging to the class A' can be interrogated, amended or newly set up as objects of the class A from the service computer 24. The expansions of the class A', in comparison to the class A, ~~can~~, cannot, however, ~~not~~ be processed by the service  
30 computer 24 until the programs in the service computer 24 have been amended at a later time in such a way that the class A' is also known in the service computer 24.

Figure 2 shows the classes A and A' and the original object a2 and the object a3. Below, the meaning of the designation "allomorphic with respect to" is explained in conjunction with ~~figure~~ Figure 2. The class A' differs from the class A only by having an additional data field 50. The data structure of the class A has  
5 therefore been expanded by the data field 50 in order to be able to take into consideration a further property of the subscribers Tln2, Tln3 during the operation of the switching office 16; for example, whether the subscriber Tln2, Tln3 is connected to the switching office 16 via an optical waveguide or via a copper conductor. For this reason, the class A' is also designated below as an expanded  
10 class A'. A data field 50' in the object a3 contains, as data item, a value which indicates that the subscriber Tln3 is connected to the switching office 16 via an optical waveguide. The object a3 is generally designated as an expanded object a3.

The class which is specified when an object is generated is designated as an original class of this object. The object a2 had, as original class, the class A  
15 indicated by an arrow 52. On the other hand, the expanded object a3 has, as original class, the expanded class A' indicated by an arrow 54.

A first possible way of defining the data structure of the class A ~~comprises~~ includes generating the class A' from the class A ~~by means of~~ via what is referred to as inheritance which is defined in object-oriented programming languages. Such  
20 programming languages are, for example, the languages C++ and CHILL. When inheritance occurs, the programmer specifies that the expanded class A' is to take over all the data structures and what are referred to as methods for processing the data structures from the class A. It is also specified that the class A' additionally contains the data field 50. Another possible way of defining the class A' is to  
25 redefine this class. In this case, the class A' is defined in the way in which it was already defined in class A. However, in addition, the data field 50 is also defined. The relationship between the corresponding parts of class A and of the expanded class A' is illustrated in ~~figure 2 by means of~~ Figure 2 via dashed lines 56.

An imaginary object a3\* contains, from the object a3, all the data fields and  
30 all the methods for processing the data fields which also would have been generated when the subscriber Tln3 is set up before the development, when the class A

existed, but the class A' did not yet exist. In the object a3\*, there is for this reason no data field corresponding to the data field 50 or 50'. This fact is indicated by dashed lines 58. The object a3\* is a visual aid for delimiting the terms "compatible with" and "allomorphic with respect to". An arrow 60 indicates that the object  
5 a3\* is compatible with the class A because it has precisely the data structures which are predefined in the class A. The expanded object a3 is on the other hand allomorphic with respect to the class A, cf. arrow 62. The object a3 has the allomorphic class A.

Allomorphy is the ability of the objects in class A' to be controlled as if they  
10 were objects of their allomorphic class A if this ability arises as a result of measures on the part of the application program. In an incremental expansion it is possible for there to be more than one allomorphic class; for example, the allomorphic class of the last expansion and the allomorphic class of the penultimate expansion.

15 An expanded object has only one allomorphic class if the expanded object is compatible in accordance with standard X.720 section 5.2.2 without the expansions to form the allomorphic class. In particular, the expanded object accordingly has all the attributes, attribute groups, control functions and confirmation methods which are also defined in the allomorphic class. Making  
20 allowance for allomorphy in the expansion of the class A ensures that the control network 10 also operates without errors after the expansion.

Figure 3 shows the processing of messages in the control computer 16 after the development of the class A into the class A', the object a2 which has been configured before the development as belonging to the class A being controlled.  
25 During the development of the class A into the class A', allomorphy was taken into account so that objects of the class A' are allomorphic with respect to the class A. In addition, in the development all the objects of the class A were converted into objects of the class A' through supplementing data fields and methods.

The control computer 16 contains an interface program 100 which processes  
30 maintenance messages coming from the service computer 24, for example a maintenance message WN1, and which transmits confirmation messages, for

example a confirmation message BN1', to the service computer 24. The interface program 100 is the interface between the service computer 24 and a plurality number of application programs in the control computer 16, of which two application programs 102 and 104 are shown in ~~figure 3~~ Figure 3. The application program 102 serves to administer the data associated with the subscribers Tln2, Tln3 connected to the switching office 16. The application program 104 is used for traffic measurement.

The maintenance message WN1 coming from the service computer is passed on, as amended maintenance message WN1', to the application program 102 when the interface program 100 is processed. On the other hand, maintenance messages which are intended for the application program 104 are passed on to the application program 104, cf. arrow 106, when the interface program 100 is processed.

After the processing of the maintenance message WN1' in the application program 102, the application program 102 generates a confirmation message BN1 for the interface 100. If the application program 104 has processed a maintenance message, it also transmits a confirmation message to the interface program 100, cf. arrow 108.

During the processing of the maintenance message WN1', the application program 102 cooperates with a database program 110 which is also present in the control computer 36 and is used to store, change, delete or read subscriber data in the memory 42. The application program 102 transmits requests in the form of messages to the database program 110; for example, the message N1. After the request in the message N1 has been executed, the database program 110 transmits a result message EN1 back to the application program 102. The application program 104 cooperates with its own database program 112 by transmitting requests to the database program 112, cf. arrow 114, and by receiving and further processing result messages from the database program 112, cf. arrow 116.

The database program 110 uses the same access method for the objects of the original class A and the objects of the developed class A'. This is possible because combination classes are used in which the data structures and methods of

the original class and of the expanded class are combined. A combination class KA is the combination of the class A and of the class A'. All the objects in the memory 42 which have the allomorphous class A as original class are expanded during the development by the additional data fields of the expanded class A. The additional  
5 data fields are filled with predefined values.

The maintenance message WN1 contains a class identifier moC which specifies the class A as the class to which the maintenance message WN1 refers. An object identifier moI specifies the object a2 to which the maintenance message WN1 refers. The maintenance message WN1 is transmitted by the service  
10 computer 24 in order to find out the subscriber data of the subscriber Tln2. All that is known in the service computer 24 is that this subscriber data is contained in the object a2 which is stored in the control computer 36. The maintenance message WN1 contains further data fields which are not illustrated. The read operation to be carried out, for example, is defined in one of these data fields.

15 When the maintenance message WN1 is processed in the interface program 100, the class A specified in the class identifier moC is determined. When the interface program 100 is processed, the class A' is determined as an alternative class ~~by means~~ via a table T by reference to the class A which is determined in this way, and the class A' is entered into the class identifier moC of the maintenance  
20 message WN1. The table T is stored in a memory 122 of the control computer 36.

The maintenance message WN1' relates, with the class identifier moC=A', to the expanded class A'. The value of the object identifier moI=a2 remains unchanged in the maintenance message WN1'. An identifier allo in the maintenance message WN1' specifies all the classes which are allomorphous with  
25 respect to the class A', i.e. in the exemplary embodiment the class A. The interface program 100 also obtains these classes from the table T. The other data of the maintenance message WN1 are transferred into the maintenance message WN1'. When the maintenance message WN1' belonging to the maintenance message WN1 is generated, the interface program 100 also carries out protocol adaptation of a  
30 transmission protocol on the transmission link 22 into a message protocol which is used within the control computer 36.

5 The message N1 which is generated by the application program 102 during the processing of the maintenance message WN1' contains an instruction B1 which specifies that data is to be read. As a parameter of the instruction B1, the message N1 contains the class A' to which the data to be read belongs, as well as the object a2 whose data is to be read. The application program 102 processes exclusively messages which refer to objects of the class A'. No further measures are taken in the application program 102 with respect to the class A.

10 The database program 110 accesses the memory 42 during the processing of the message N1 in order to read the data of the subscriber Tln2 which is stored in the object a2. The object a2 also contains an origin identifier oC in which the class which has been specified when the object a2 was generated is specified. The database program 110 reads the data requested with the instruction B1 in the object a2 and enters this data into the result message EN1. In addition, in the result message EN1 there is a note, ~~by means of~~ via a response identifier AB1, that the  
15 result message EN1 contains the result which has been generated when the instruction B1 was processed.

Furthermore, in the result message EN1 the class A' is specified as the class which has been affected by the result message EN1. The origin identifier oC=A is also transferred from the database program 110 in the result message EN1 to the  
20 application program 102.

When the application program 102 is processed it is a prerequisite that all the messages to be processed refer to the class A' and not to the class A. The value of the origin identifier oC=A is transferred as a value of the class identifier moC in the confirmation message BN1 by the application program 102. For this value  
25 assignment it is not necessary for the application program 102 to be able to process objects of the class A. In addition, the confirmation message BN1 contains the origin identifier oC=A and the interrogated subscriber data of the subscriber Tln2. The interface program 100 does not need to remember the class identifier of the maintenance message WN1 by virtue of this procedure.

30 The confirmation message BN1 is processed by the interface program 100 and transmitted, as confirmation message BN1', to the service computer 24 on the

transmission link 22 in accordance with the transmission protocol. By reference to the table T, the interface program 100 determines which data fields in the message BN1 are not contained in objects of the class A. These data fields are not transferred into the confirmation message BN1'.

5           The service computer 24 receives the confirmation message BN1' and can process it as a message which refers to an object of the class A. Objects of the class A' in the control computer 36 are controlled from the service computer 24 as if they were objects of the class A. The development in the control computer 36 does not, therefore, restrict the operating features of the service computer 24.

10           Figure 4 shows the processing of messages in the control computer 16 after the development of the class A into the class A', the object a3 which is not generated until after the development being controlled.

~~By means of~~ Via inputs by an operator into the service computer 24, a maintenance message WN2 is generated in the service computer 24, said the  
15 maintenance message WN2 having the function of generating in the control computer 16 the object a3 for the subscriber data of the subscriber Tln3. For this reason, the maintenance message WN2 contains a coding for the instruction "Generate" in an instruction field. The class identifier moC in the maintenance message WN2 identifies the class A as the class to which the maintenance message  
20 WN2 refers. The object identifier moI of the maintenance message WN2 identifies the object a3 which is to be generated via the maintenance message WN2. The maintenance message WN2 also contains further data which is not illustrated.

          The interface program 100 processes the maintenance message WN2 in the same way as explained above for the maintenance message WN1. When the  
25 maintenance message WN2 is processed, the table T stored in the memory 122 is again used to determine the alternative class A' for the class A specified in the class identifier moC of the maintenance message WN2 and to use it as the class identifier moC of the maintenance message WN2'. The object identifier moI=a3 has the same value in the maintenance message WN2' as in the maintenance message  
30 WN2. The remaining data is also transferred from the maintenance message WN2 into the maintenance message WN2' when the interface program 100 is executed.

In addition, in the identifier allo of the maintenance message WN2', it is noted that the class A is the class with respect to which the class A' is allomorphous. The maintenance message WN2' generated in this way is transmitted from the interface program 100 to the application program 102 in accordance with the protocol in the control computer 36.

When the maintenance message WN2' is processed, the application program 102 transmits a message N2 to the database program 110 in order to allow the object a3 to be created in the memory 42. The message N2 contains, in encrypted form, the instruction "Generate", the expanded class A' and the name of the object a3 to be created.

When the message N2 is processed, the object a3 is generated in the memory 42 ~~by means of~~ via the database program 110 by assigning to this object specific memory cells which are filled with initial values. In the origin identifier oC of the object a3 there is a note of the class A' because this class was specified when the object a3' was generated. An identifier allo in the object a3 refers to the class A because the object a3 is allomorphous with respect to the class A.

The database program 110 generates a result message EN2 in order to confirm the generation of the object a3. The result message EN2 contains a response identifier AB2 which specifies that the result message EN2 was produced when an instruction "Generate" was produced. In addition, the result message EN2 contains an identifier which refers to the class A', the origin identifier oC=A', the identifier allo=A and further data (not illustrated) of the object a3.

The result message EN2 is processed by the application program 102, a confirmation message BN2 being generated. The value of the origin identifier oC=A' is used as a value of the class identifier moC in the confirmation message BN2. The other data items in the result message EN2 are transferred into the confirmation message BN2.

When the interface program 100 is processed, after the reception of the confirmation message BN2, a confirmation message BN2' is generated in accordance with the transmission protocol used on the transmission link 22. The confirmation message BN2' contains all the data of the confirmation message BN2



because the interface program 100 determines, by reference to the table T, that it is not necessary to remove any data fields if the confirmation message BN2 has the class A' as class identifier.

During the processing of the confirmation message BN2' in the service  
5 computer 24, it is determined by reference to the identifier allo that the confirmation message BN2' relates to data of an object of the class A or that the confirmation message BN2' can be treated as if it contained data of an object of the class A. Although the program in the service computer 24 has not been changed after development in the control computer 36, objects of the class A' in the control  
10 computer 36 can be controlled from the control computer 24 as if they were objects of the class A. As long as only the class A is known in the service computer A, it is also the case that only the data items contained in objects of the class A are processed in the confirmation message BN2'.

~~By means of~~ Via the method explained with reference to ~~figures 3~~ Figures 3  
15 and 4, it is also possible to support classes which are allomorphic with respect to a plurality number of classes. If, for example, the class A' is developed into a class A'', objects can be allomorphic with respect to the classes A' and A. The control computer 36 ~~can~~ then can be executed by service computers 24 during whose programming at least one of the classes A, A' or A'' was known. It is possible to  
20 dispense with the class A as soon as all the service computers know at least the class A'.

The method explained with reference to ~~figures 3~~ Figures 3 and 4 is applied in maintenance messages for generating objects, and in maintenance messages for reading objects. In addition, this method can be used to change data in objects and  
25 delete objects. For these methods to operate without errors, the service computer 24 must be capable:

- of assigning the confirmation messages BN to the associated maintenance messages WN,
- of reading and evaluating the identifier allo in the confirmation  
30 messages BN,

- of skipping over unknown name associations (cf. explanations relating to ~~fig-5~~ Fig. 5) and unknown identifier values and parameters,
- of storing the origin class of an object if there is a database in the service computer,
- of skipping over unknown optional values, and
- of skipping over unknown values of the "enumerated" data type.

In other words, the service computer 24 must be programmed in such a way that objects can be transferred from it into a control computer with a greater degree of knowledge.

The interface program 100 has the following features:

- When a class is expanded it is necessary to incorporate the expanded class into the table T, ~~which means~~ such that for each expanded class it is necessary to store the classes with respect to which the expanded class is allomorphous.
- The name associations which have been made by the expanded class must be stored in the table T.
- The interface program 100 is to remove from the confirmation messages BN coming from the application program 102 the data which does not belong to the compatible class if the confirmation message BN transmitted to the service computer 24 does not relate to the non-expanded class.
- The interface program 100 must be able to process filter instructions. This is explained in more detail below with reference to ~~figure~~ Figure 5.

The application program 102 fulfills the following requirements:

- As soon as the system moves over to the expanded class, only objects of the new class are then generated, irrespective of the knowledge of the service computer.

- The new classes are either expanded classes or classes which have nothing to do with the previous classes.

The database program 110 is configured in such a way that, after a development, the entire data stock relating to the origin class is converted into a data stock of the expanded class.

The creation of the table T and the conversion of the data stock in the database of the database program 110 are supported by service programs. These service programs evaluate description languages which are used to specify the expansion of classes.

Conditions which have to be complied with when taking into consideration allomorphy are dealt with below. These conditions apply on the level of the classes although, according to standard X.720, allomorphy is firstly a property of an object. In the standard X.720, the following conditions are mentioned:

- conditions for the expanded class in section 5.2.2.1,
- conditions for what are referred to as program packages in the sections 5.2.2.1 and 5.2.2.2,
- conditions for identifier values in the sections 5.2.2.3 and 5.3.4.1,
- conditions for what are referred to as identifier groups in section 5.2.2.4,
- conditions for actions, confirmations and parameters in section 5.2.2.54,
- conditions for the behavior of the objects in section 5.2.2.6, and
- conditions for the name association in section 5.3.4.1.

Figure 5 shows, in a part a, a section of what is referred to as a binary tree which defines the name association of objects of the class A and A'. Name association is the assignment of the object to what is referred to as a superordinate object, said the assignment being used for the definition of a uniquely defined name for an object. Identical object names for different objects can be used if the objects each belong to another superordinate object. A name which is uniquely defined in

the control computer 16 is then produced from the name of the superordinate object and the name of the objects which are subordinated in this way. The superordinate object is to be specified when an object is generated. The subordinate object is considered as being contained in the superordinate object, referred to in English as  
5 "containment".

An object b1 of the class B is the superordinate object for the objects a2 and a3, which both belong to the class A' after the development, but have different original classes oC=A and oC=A'. The name associations are noted in the table T which is used by the interface program 100.

10 For the portion of the name tree shown in part A of ~~figure~~ Figure 5 there is a note in table T that objects of the class B contain subordinate objects of the original class A and of the class A'. At each development, the name associations in the table T are also adapted to the name tree which is also decisive after the development.

15 In part b of ~~figure~~ Figure 5, an access to objects of both classes A and A' is illustrated after the service computer 24 also knows both classes A and A'. A maintenance message WN3 contains a filter instruction  $F1=((oC=A) \text{ OR } (oC=A'))$  in which it is defined that subordinate objects of the class A or A' are to be registered. The class identifier moC=B of the maintenance message WN3 specifies  
20 that the maintenance message WN3 also refers to the class B. The object identifier moI=b1 of the maintenance message WN3 specifies that the object b1 is to be processed; i.e., is the superordinate object.

When the interface program 100 is executed in the switching office 16, the filter instruction F1 is transmitted unchanged to the application program 102. This  
25 means As such in particular, that operations which refer to the original class A do not have to be replaced in the filter instruction F1 by operations which refer to the expanded class A'. This measure ensures that the service computer 24 can distinguish between objects of the original class A and of the original class A'.

The application program 102 causes the data of the objects a2 and a3 whose  
30 original class is the class A and A', respectively, to be read from the memory 42 ~~by means of~~ via database program 110. Two confirmation messages BN3 and BN4

(not illustrated) are transmitted to the interface program 100. A new confirmation message BN3', which contains the class identifier moC=A, the object identifier moI=a2, the origin identifier oC=A and the auxiliary identifier allo={} is generated from the confirmation message BN3 in the interface program 100. A confirmation  
5 message BN4', which contains the class identifier moC=A', the object identifier moI=a3, the origin identifier oC=A' and the auxiliary identifier allo={A} is generated from the confirmation message BN4 in the interface program 100.

The CCITT standard X.734 "Information Technology - Open Systems Interconnection - Systems Management: Event Report Management Function" from  
10 1993 explains event control in the control network 10, cf. figure 1. What are referred to as discriminators are used which pass on events to the service computer 24 within the switching office 16 only under certain conditions. After the development of the class A into the class A', it is sufficient to convert the discriminators which relate to the class A into discriminators which relate to the  
15 class A'. If new discriminators are generated after the development, the class A is replaced by the class A' if the class A is specified as a selection criterion for the passing on of the messages.

Although the present invention has been described with reference to specific embodiments, those of skill in the art will recognize that changes may be made  
20 thereto without departing from the spirit and scope of the invention as set forth in the hereafter appended claims.

Abstract

ABSTRACT OF THE DISCLOSURE

Method and network element for operating a telecommunications network

- A method for operating a telecommunications network is explained. A
- 5 wherein a network element (16) at a network node of the telecommunications network is controlled by a control computer(36). ~~The, the~~ control computer (16) is maintained from a service computer(24). ~~Allomorphy, allomorphy~~ is taken into account in the development of application programs (102, 104) so that even a service computer (24) which is not developed can maintain the control
- 10 computer(36). ~~Making, and making~~ allowance for allomorphy results in little additional expenditure because an interface program (100) is used in which essential processing steps are carried out for all the application programs(102, 104), ~~said, the~~ processing steps ensuring that allomorphy is taken into consideration.

- 15 Figure-1

BOX PCT  
IN THE UNITED STATES ELECTED/DESIGNATED OFFICE  
OF THE UNITED STATES PATENT AND TRADEMARK OFFICE  
UNDER THE PATENT COOPERATION TREATY-CHAPTER II

**SUBMISSION OF DRAWINGS**

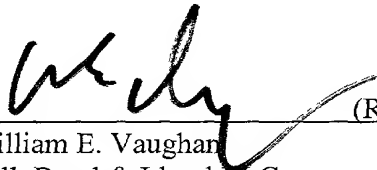
APPLICANTS: Maria Froeschl et al.. DOCKET NO.: 112740-320  
SERIAL NO: GROUP ART UNIT:  
FILED: EXAMINER:  
INTERNATIONAL APPLICATION NO. PCT/EP00/02070  
INTERNATIONAL FILING DATE: 09 March 2000  
INVENTION: METHOD AND NETWORK ELEMENT FOR OPERATING A  
TELECOMMUNICATIONS NETWORK

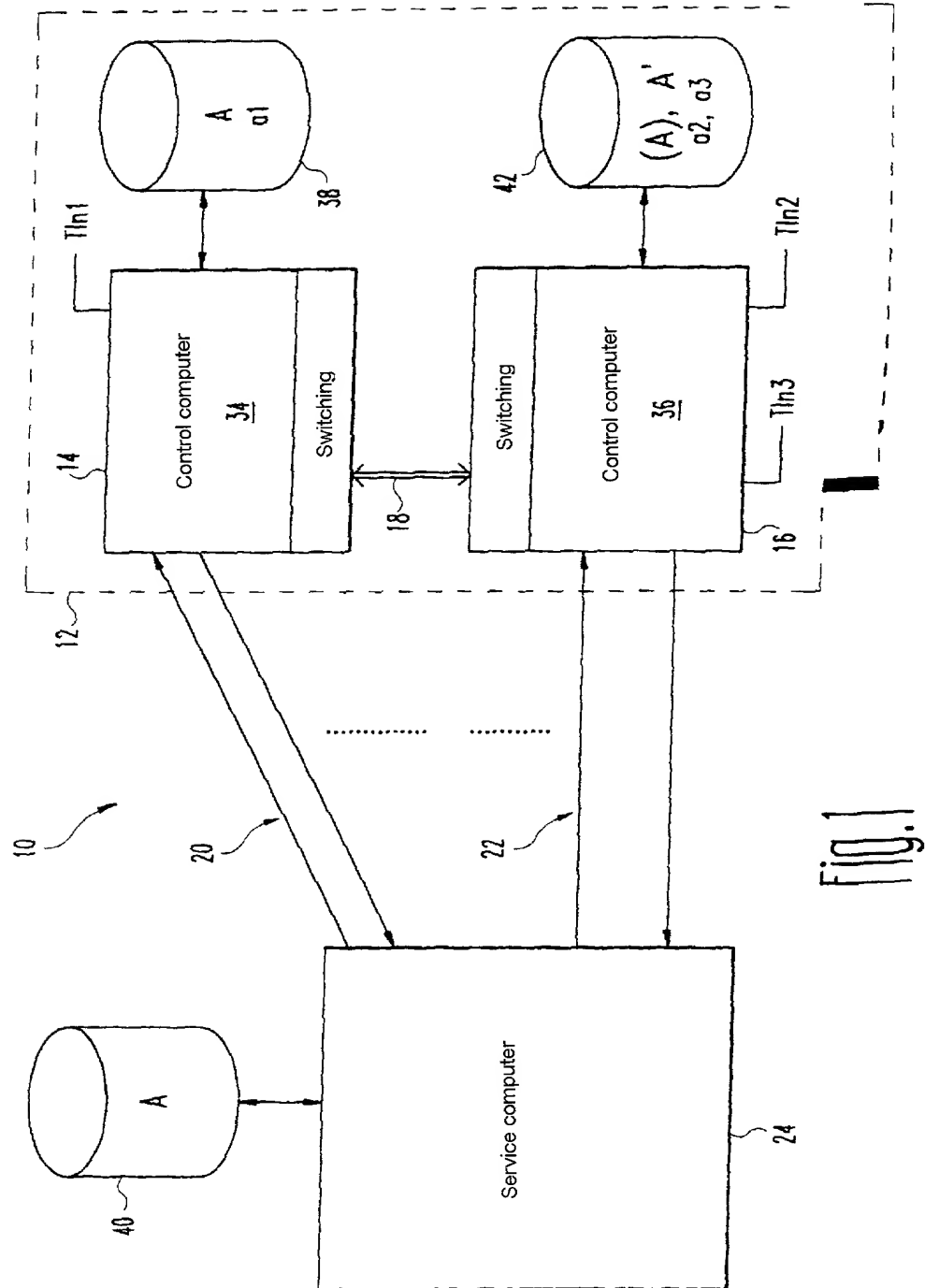
Assistant Commissioner for Patents,  
Washington, D.C. 20231

Sir:

Applicants herewith submit five sheets (Figs. 1-5) of drawings for the above-  
referenced PCT application.

Respectfully submitted,

  
(Reg. No. 39,056)  
William E. Vaughan  
Bell, Boyd & Lloyd LLC  
P.O. Box 1135  
Chicago, Illinois 60690-1135  
(312) 807-4292  
Attorneys for Applicants



11.



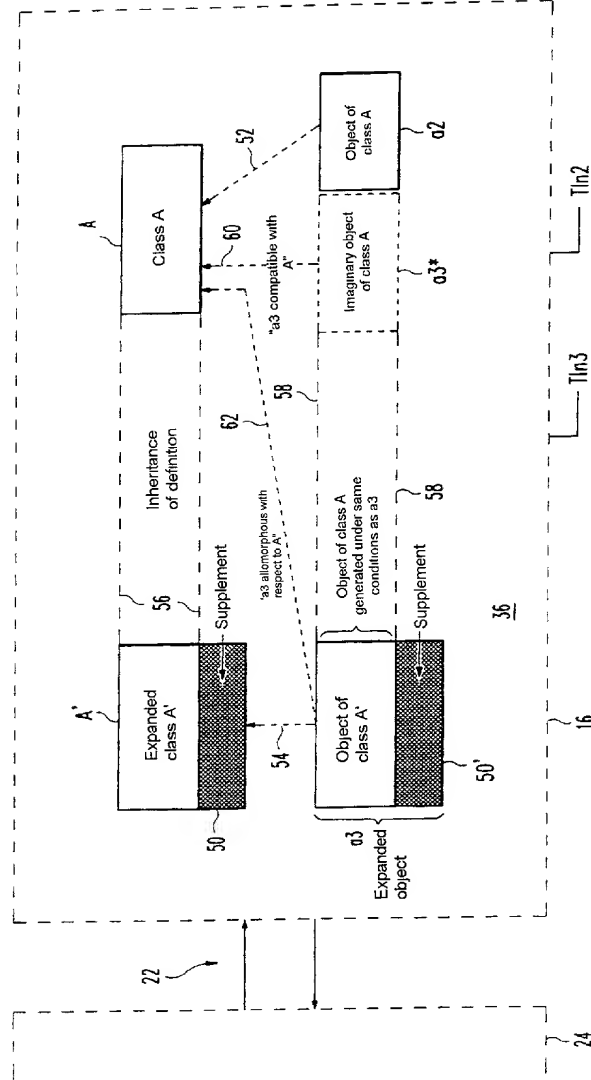


Fig. 2

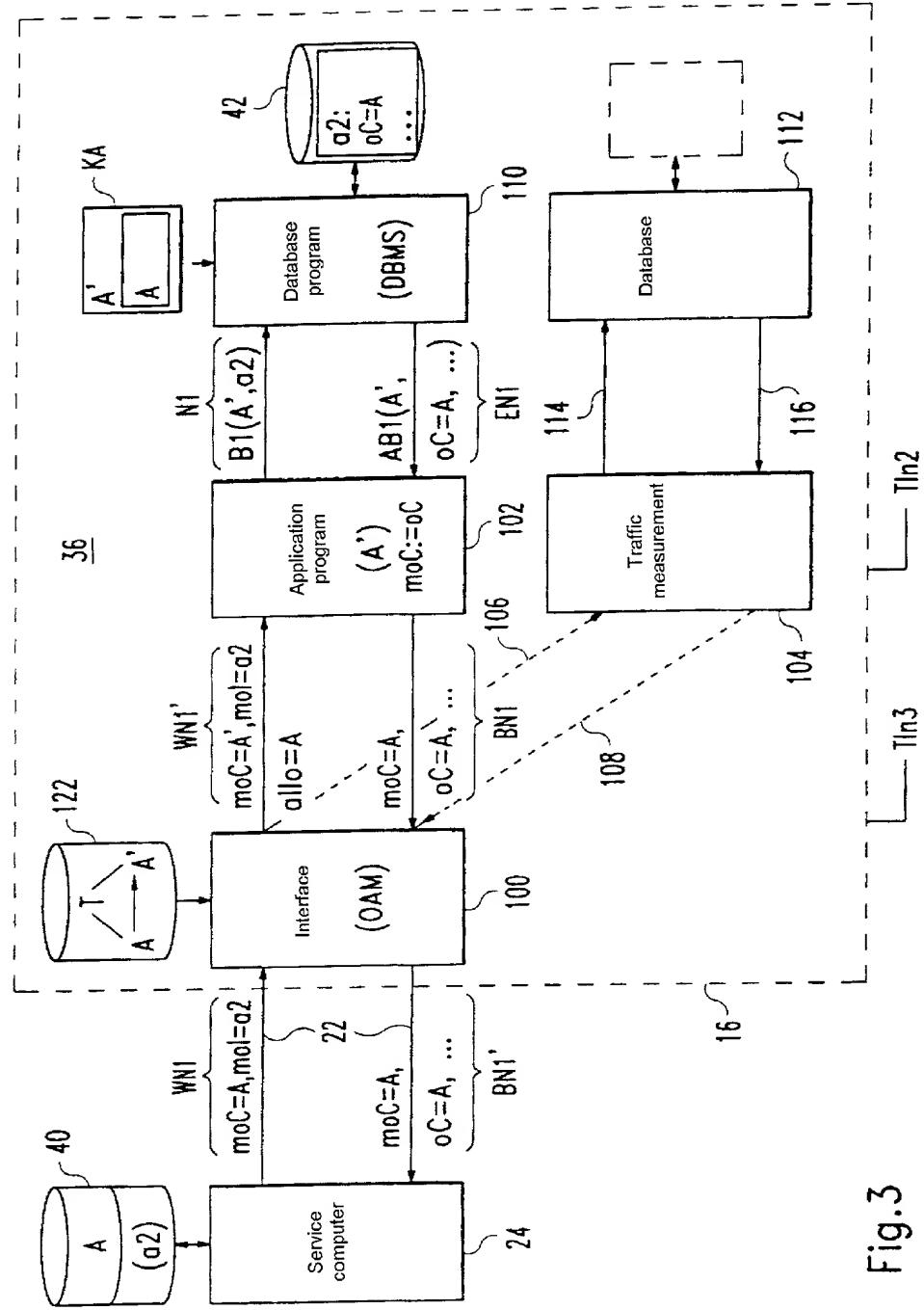


Fig. 3

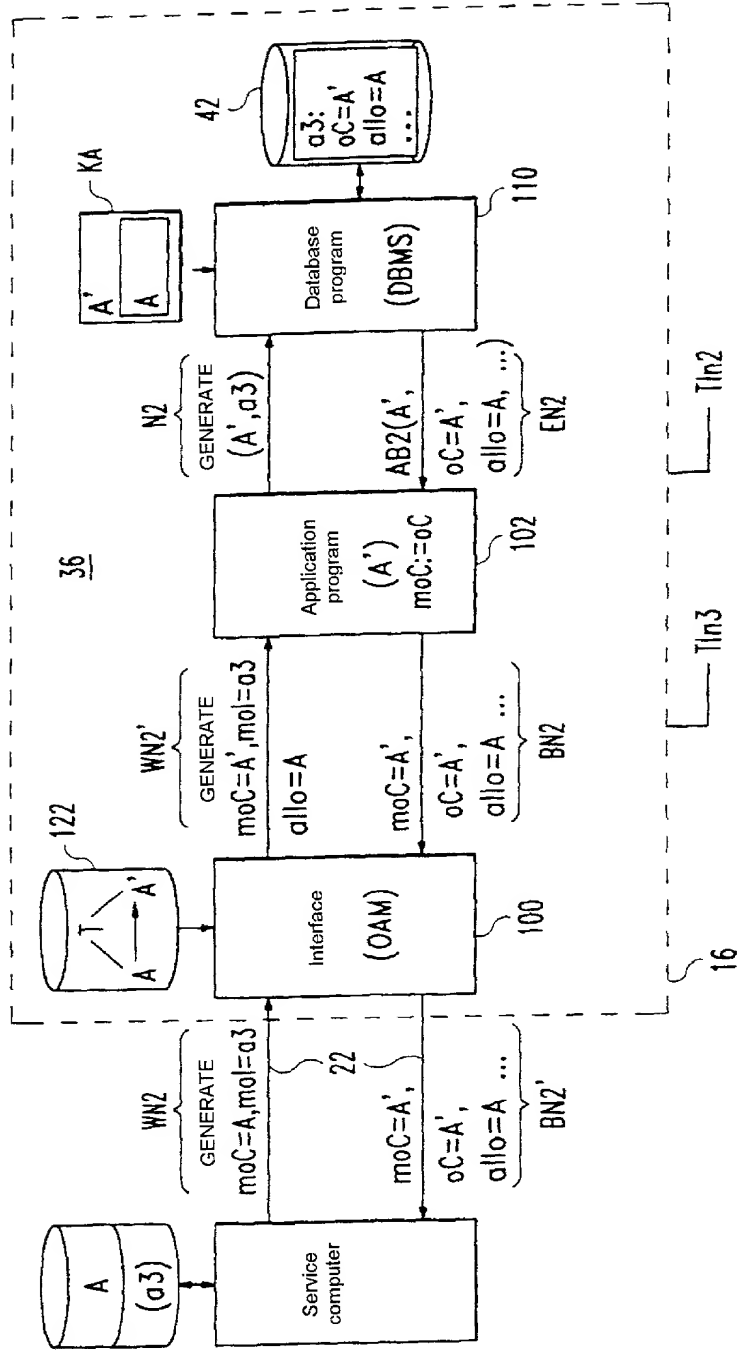
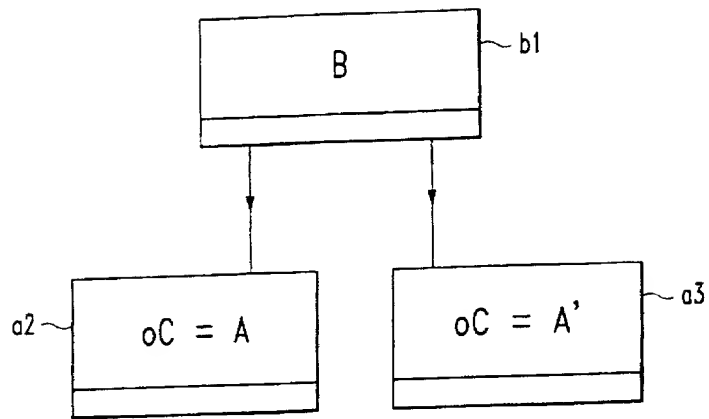
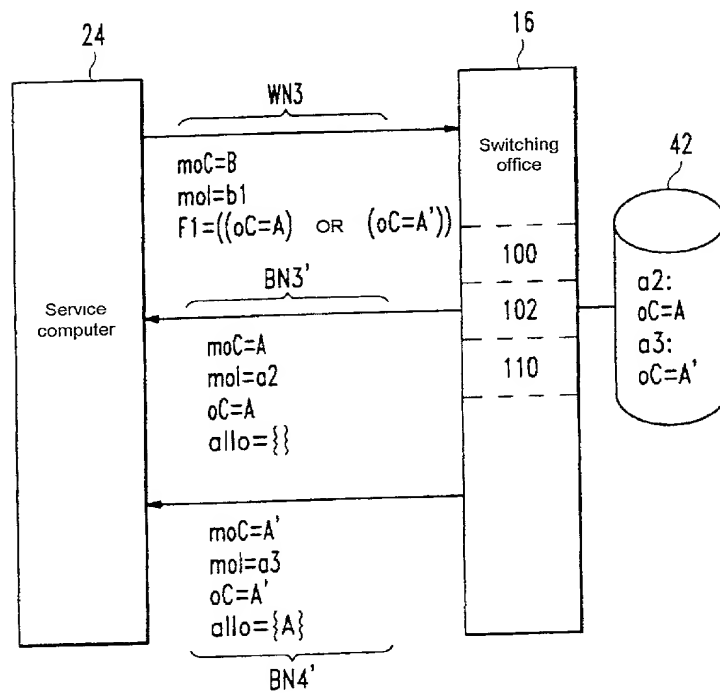


Fig. 4



a)



b)

Fig.5

GR 99 P 1391

## Description

Method and network element for operating a telecommunications network

5

The invention relates to a method for operating a telecommunications network, in which a network element at a network node of the telecommunications network is controlled by a control computer. The network element is, for example, a switching office for switching links, what is referred to as a cross-connector or a concentrator unit for connecting a plurality of subscribers to a connecting line. In addition to the operating system for operating the control computer, a plurality of application programs, during whose execution application objects are processed, are stored in the control computer. The application objects are associated with data with a predefined data structure and preferably also predefined methods for processing the data. The data structure and the methods are dependent on a class which is also to be specified when the respective application object is generated. A link via which the control computer is maintained by means of maintenance messages is set up between a service computer and a control computer.

Such methods are used for controlling the telecommunications network if, for example, a new switching device is to be put into operation as a network element or if subscriber data in the switching device have to be changed at a later date, such as is the case when new subscribers are connected or when an existing subscriber moves house. Efficient methods for controlling the telecommunications network are obtained if what are referred to as open systems are used which are programmed in compliance with standards which apply worldwide. For example, standards of the ISO (International Standardization Organization) and of the ITU (International Telecommunication Union) with its



of such open systems. A separate control network is to be used to control the telecommunications network. The interfaces between the service computer and switching device are standardized in protocols Q1, Q2 and Q3.

5

The application objects are defined as objects of an object-oriented language, for example in the language C++ or CHILL. If the application programs are developed, it is necessary to ensure that the control  
10 network also operates without faults with the new application programs. This means in particular that application objects which are considered by the service computer as belonging to an original class cannot readily be assigned to an amended alternative class.

15

This problem is mentioned in the CCITT standard X.720 (01/92) - "Information Technology - Open Systems Interconnection - Structure of Management Information: Management Information Model" - in the section 5.2.1.  
20 In the section 5.2.3 of the standard X.720, two methods for solving the problem are proposed. In the first method, on the part of the application program, a programming technique is used which takes allomorphy into consideration. Allomorphy is the ability of a  
25 specific application object of the alternative class to be controlled as if it were an object of the original class if this ability arises as a result of measures on the part of the application program. The other method comprises taking measures on the part of the service  
30 computer which permit cooperation between the service computer and the application program even when the application program is developed.

European patent application EP-A-0 817 422 discloses a  
35 method for implementing controlled objects in a subsystem of a controlled system in a network, at least

one control system and one controlled system being present. The controlled objects are implemented independently of other subsystems without knowing the type of the objects in the other subsystems. They can  
5 be connected to other objects and can transmit messages to them. For this purpose, a first object is generated for cooperating with an abstract object. The abstract object has a defined interface which is called up using the first object and which inherits the abstract object  
10 from a second object, the generated second object being unknown to the first, and the second object being intended for cooperation with the first object. The first object which cooperates with the second object considers the second object as an object of the  
15 abstract type. However, in this method known from EP-A-0 817 422, there is no use of programming technology which takes into consideration the allomorphy in accordance with the section 5.2.3 of the standard X.720.

20 The object of the invention is to disclose a simple method for operating a telecommunications network in which allomorphy is taken into consideration.



This object is achieved by means of a method having the features of patent claim 1. Developments are disclosed in the subclaims.

5 In the method according to the invention, in each case a class identifier in which the class to which the maintenance message relates is specified is determined from the maintenance messages during the processing of an interface program which is used for a plurality of  
10 application programs. The class identifier in the maintenance message specifies the class, known in the service computer, of an application object to be processed. As a result of the development, the class known in the service computer sometimes deviates from  
15 the actual class of the application object. When the interface program is processed, an alternative identifier is determined by reference to the class identifier, said alternative identifier specifying an alternative class to which the application object in  
20 the network element is assigned. The alternative identifier is incorporated into an amended maintenance message. When the amended maintenance message is processed by an application program, the application object is then processed as belonging to the  
25 alternative class. This is possible because the application object is allomorphous with respect to the class which is known in the service computer and which is required for the application object in the unamended maintenance message by the service computer.

30 The interface program performs the assignment of the alternative identifiers to the class identifiers centrally for a plurality of application programs. In the method according to the invention, this step does  
35 not have to be programmed in each application program but rather only once in the interface program. Given several hundred application programs per control computer, this reduces the complexity of programming, maintenance and documentation considerably. The

application programs are kept free of additional steps which are necessary when taking allomorphy into consideration because these steps are carried out centrally in the interface program which is situated upstream.

Some of the additional steps are also carried out in databases which are situated downstream and which are utilized by the application programs.

5 The execution of the assignment of alternative identifiers and auxiliary identifiers in a central interface program is possible because allomorphy is defined at the class level in the method according to the invention. Such a definition is not mentioned in  
10 the standard X.720 but is nevertheless compatible with the standard. Allomorphy at the class level means that all the objects of the alternative class can be carried out as if they were objects of the original class. A definition of allomorphy which is referred to all the  
15 objects of the alternative class then does not result in any disadvantages if predefined programming rules are complied with. Examples of such programming rules are explained below in relation to the exemplary embodiments.

20 The method according to the invention makes it possible to comply easily with the stipulations of the standard X.720. The application programs in the control computer can be developed with a small amount of additional  
25 expenditure, in which case it is always ensured that no errors occur when the control network is operated even when programs remain unchanged in the service computer.

In one development, a table with which the alternative  
30 identifiers are assigned to the class identifier is used in the interface program. The table is stored in the memory of the control computer. An entry in the table is read in that a memory cell which is assigned to the class identifier and contains the alternative  
35 identifier belonging to the class identifier is read. To determine the alternative identifier, all that is therefore required is a single read access to the memory. If the alternative identifiers change as a result of developments of the application programs, all



the contents of the table can be easily replaced or expanded.

If, in another development after the amended  
5 maintenance message has been processed, the application  
program generates a confirmation message in which the  
class specified during the generation of the  
application object to be processed is specified as a  
class identifier, the confirmation message can  
10 subsequently easily be processed by the interface  
program. For example, when the interface program is  
processed, it is possible to determine by reference to  
the class identifier which data are to be removed from  
the confirmation message. To do this, the table used in  
15 the interface program is expanded to such an extent  
that entries to the permitted data are also associated  
with each class identifier. The interface program then  
generates from the confirmation message a new  
confirmation message which contains only such data of  
20 an application object of the class to which the  
confirmation message refers.

The class specified when the application object to be  
processed is generated is stored in one development as  
25 an origin class in the data of the application object  
to be processed. When the application program is  
processed, the origin class is then used as a class  
identifier. This procedure makes the origin class  
available easily.

30

If, in another development, the confirmation message  
also contains an auxiliary identifier in which at least  
one class is designated which is known in the service  
computer and/or in at least one other service computer  
35 as the class to which the application object to be  
processed belongs, the program in the service computer  
can determine later, by reference to the auxiliary  
identifier, how the received confirmation message is to  
be processed. This is significant in particular if the

class specified in the class identifier of the confirmation message received by the service computer is not yet known in the service computer. The service computer then determines the class to which the confirmation message refers by reference to the class or classes specified in the auxiliary identifier. The auxiliary identifier contains, in other words, the classes with respect to which the application object is allomorphous. If the confirmation message contains not only the class identifier but also an origin identifier in which the origin class is specified, the requirements of the protocol for the exchange of messages in the control computer and also for the protocol for the exchange of messages between the service computer and control computer can be fulfilled.

In another development, at least one class which is known in the service computer and/or in at least one other service computer for the application object is stored as an allomorph class in the data of the application object. When the application program is processed, the allomorph class is then used as an auxiliary identifier. This measure results in an easy-to-manage data structure in which the application objects administer their allomorph classes themselves. It is not necessary to take any additional measures with respect to the allomorph class in the interface program or in the application program.

In another development, the interface program is also responsible for other interface functions. For example, for the control of events for defining the processing sequence of the maintenance messages or for protocol adaptations of these messages, referred to in English as "basic encoding". As a result of this measure, on the control computer there is just a single interface program which is programmed and maintained in a standardized fashion.

The exemplary embodiments of the invention are explained below with reference to the drawings, in which:

Figure 1 shows part of a control network for controlling a telecommunications network,

5 Figure 2 shows the development of an original class A into an expanded class A', whose objects can be controlled as objects of the old class A during the operation of the control network,

10 Figure 3 shows the processing of messages in the control computer of a switching unit after the development, an object being controlled which has been generated before the development,

15 Figure 4 shows the processing of messages in the control computer after the development, an object being controlled which has been generated after the development,

20 Figure 5 shows the association of the names of classes A and A', and an access to objects of the two classes by means of a filter function.

Figure 1 shows part of a control network 10 for  
25 controlling a telecommunications network 12. The telecommunications network 12 contains a multiplicity of switching offices, of which the switching offices 14 and 16 are illustrated in figure 1. The telecommunications network 12 also includes connecting  
30 lines between the switching offices, of which a connecting line 18 between the switching office 14 and the switching office 16 is illustrated in figure 1. The telecommunications network 12 connects the subscribers of the telecommunications network 12, for example a  
35 subscriber Tln1 connected to the switching office 14 and a subscriber Tln2 connected to the switching office 16.





link 20 transmits maintenance messages from the service computer 24 to the switching office 14 in order, for example, to change subscriber data of the subscriber Tln1 in the switching office 14. The switching office 5 14 itself transmits confirmation messages to the service computer 24 in order to signal satisfactory processing of the received maintenance message. The transmission link 22 serves for bidirectional transmission of data between the service computer 24 10 and switching office 16.

The maintenance messages are processed in the switching office 14 by a control computer 34 and in the switching office 16 by a control computer 36. The data structures 15 to which the maintenance messages refer belong to the same class A in the service computer 24 and in the switching office 14. The switching office 16 on the other hand contains data structures of a class A', which have been developed in comparison with the class 20 A required in the service computer 24. Error-free operation of the control network 10 is ensured with respect to the switching office 16 by virtue of the fact that allomorphy has been taken into consideration in the development of the class A into class A'. What 25 allomorphy means in this context is explained below with reference to figure 2.

In the exemplary embodiment illustrated in figure 1, the class A which defines, for example, the data 30 structure of the subscriber data, for example the call number and the useable services of the telecommunications network 12, is used in the switching office 14. The subscriber data of the subscriber Tln1 is stored in an object a1 in accordance with the data 35 structure predefined by the class A in a memory 38 of the control computer 34. The class A is also known in the service computer 24, indicated by the letter A in a memory 40 of the service computer 24.

[illegible]

subscriber data of the subscriber Tln2. The object a2 was firstly stored in the memory 42 before the class A was developed into the class A'. However, in the development, the original object a2 was converted, specifically into an expanded object a2 of the class A' by supplementing a data field. An object a3 in the memory 42 belongs to the class A' and contains the subscriber data of a subscriber Tln3 whose connection was not set up in the switching office 16 until after the development. Although the programs in the service computer 24 only support objects of the class A, the objects a2 and a3 belonging to the class A' can be interrogated, amended or newly set up as objects of the class A from the service computer 24. The expansions of the class A', in comparison to the class A, can, however, not be processed by the service computer 24 until the programs in the service computer 24 have been amended at a later time in such a way that the class A' is also known in the service computer 24.

Figure 2 shows the classes A and A' and the original object a2 and the object a3. Below, the meaning of the designation "allomorphous with respect to" is explained in conjunction with figure 2. The class A' differs from the class A only by having an additional data field 50. The data structure of the class A has therefore been expanded by the data field 50 in order to be able to take into consideration a further property of the subscribers Tln2, Tln3 during the operation of the switching office 16, for example whether the subscriber Tln2, Tln3 is connected to the switching office 16 via an optical waveguide or via a copper conductor. For this reason, the class A' is also designated below as an expanded class A'. A data field 50' in the object a3 contains, as data item, a value which indicates that the subscriber Tln3 is connected to the switching office 16 by means of an optical waveguide. The object a3 is generally designated as an expanded object a3.

The class which is specified when an object is generated is designated as an original class of this object. The object a2 had, as original class, the class A indicated by an arrow 52. On the other hand, the  
5 expanded object a3 has, as original class, the expanded class A' indicated by an arrow 54.

A first possible way of defining the data structure of the class A comprises generating the class A' from the  
10 class A by means of what is referred to as inheritance which is defined in object-oriented programming languages. Such programming languages are, for example, the languages C++ and CHILL. When inheritance occurs, the programmer specifies that the expanded class A' is  
15 to take over all the data structures and what are referred to as methods for processing the data structures from the class A. It is also specified that the class A' additionally contains the data field 50. Another possible way of defining the class A' is to  
20 redefine this class. In this case, the class A' is defined in the way in which it was already defined in class A. However, in addition, the data field 50 is also defined. The relationship between the corresponding parts of class A and of the expanded  
25 class A' is illustrated in figure 2 by means of dashed lines 56.

An imaginary object a3\* contains, from the object a3, all the data fields and all the methods for processing  
30 the data fields which also would have been generated when the subscriber Tln3 is set up before the development, when the class A existed, but the class A' did not yet exist. In the object a3\*, there is for this reason no data field corresponding to the data field 50 or 50'. This fact is indicated by dashed lines 58. The  
35 object a3\* is a visual aid for delimiting the terms "compatible with" and "allomorphic with respect to". An arrow 60 indicates that the object a3\* is compatible with the class A because it has precisely the data

099544-01391

structures which are predefined in the class A. The expanded object a3 is

get "hsghh"

on the other hand allomorphous with respect to the class A, cf. arrow 62. The object a3 has the allomorphous class A.

- 5 Allomorphy is the ability of the objects in class A' to be controlled as if they were objects of their allomorphous class A if this ability arises as a result of measures on the part of the application program. In an incremental expansion it is possible for there to be  
10 more than one allomorphous class, for example the allomorphous class of the last expansion and the allomorphous class of the penultimate expansion.

- 15 An expanded object has only one allomorphous class if the expanded object is compatible in accordance with standard X.720 section 5.2.2 without the expansions to form the allomorphous class. In particular, the expanded object accordingly has all the attributes, attribute groups, control functions and confirmation  
20 methods which are also defined in the allomorphous class. Making allowance for allomorphy in the expansion of the class A ensures that the control network 10 also operates without errors after the expansion.

- 25 Figure 3 shows the processing of messages in the control computer 16 after the development of the class A into the class A', the object a2 which has been configured before the development as belonging to the class A being controlled. During the development of the  
30 class A into the class A', allomorphy was taken into account so that objects of the class A' are allomorphous with respect to the class A. In addition, in the development all the objects of the class A were converted into objects of the class A' through  
35 supplementing data fields and methods.

The control computer 16 contains an interface program 100 which processes maintenance messages coming from the service computer 24, for example a maintenance





service computer 24 and a plurality of application programs in the control computer 16, of which two application programs 102 and 104 are shown in figure 3. The application program 102 serves to administer the data associated with the subscribers Tln2, Tln3 connected to the switching office 16. The application program 104 is used for traffic measurement.

The maintenance message WN1 coming from the service computer is passed on, as amended maintenance message WN1', to the application program 102 when the interface program 100 is processed. On the other hand, maintenance messages which are intended for the application program 104 are passed on to the application program 104, cf. arrow 106, when the interface program 100 is processed.

After the processing of the maintenance message WN1' in the application program 102, the application program 102 generates a confirmation message BN1 for the interface 100. If the application program 104 has processed a maintenance message, it also transmits a confirmation message to the interface program 100, cf. arrow 108.

During the processing of the maintenance message WN1', the application program 102 cooperates with a database program 110 which is also present in the control computer 36 and is used to store, change, delete or read subscriber data in the memory 42. The application program 102 transmits requests in the form of messages to the database program 110, for example the message N1. After the request in the message N1 has been executed, the database program 110 transmits a result message EN1 back to the application program 102. The application program 104 cooperates with its own database program 112 by transmitting requests to the database program 112, cf. arrow 114, and by receiving

and further processing result messages from the  
database program 112, cf. arrow 116.

added in 1999

The database program 110 uses the same access method for the objects of the original class A and the objects of the developed class A'. This is possible because combination classes are used in which the data structures and methods of the original class and of the expanded class are combined. A combination class KA is the combination of the class A and of the class A'. All the objects in the memory 42 which have the allomorphous class A as original class are expanded during the development by the additional data fields of the expanded class A. The additional data fields are filled with predefined values.

The maintenance message WN1 contains a class identifier moC which specifies the class A as the class to which the maintenance message WN1 refers. An object identifier moI specifies the object a2 to which the maintenance message WN1 refers. The maintenance message WN1 is transmitted by the service computer 24 in order to find out the subscriber data of the subscriber Tln2. All that is known in the service computer 24 is that this subscriber data is contained in the object a2 which is stored in the control computer 36. The maintenance message WN1 contains further data fields which are not illustrated. The read operation to be carried out, for example, is defined in one of these data fields.

When the maintenance message WN1 is processed in the interface program 100, the class A specified in the class identifier moC is determined. When the interface program 100 is processed, the class A' is determined as an alternative class by means of a table T by reference to the class A which is determined in this way, and said class A' is entered into the class identifier moC of the maintenance message WN1. The table T is stored in a memory 122 of the control computer 36.

[illegible]

identifier moI=a2 remains unchanged in the maintenance message WN1'. An identifier allo in the maintenance message WN1' specifies all the classes which are allomorphous with respect to the class A', i.e. in the  
5 exemplary embodiment the class A. The interface program 100 also obtains these classes from the table T. The other data of the maintenance message WN1 are transferred into the maintenance message WN1'. When the maintenance message WN1' belonging to the maintenance  
10 message WN1 is generated, the interface program 100 also carries out protocol adaptation of a transmission protocol on the transmission link 22 into a message protocol which is used within the control computer 36.

15 The message N1 which is generated by the application program 102 during the processing of the maintenance message WN1' contains an instruction B1 which specifies that data is to be read. As a parameter of the instruction B1, the message N1 contains the class A' to  
20 which the data to be read belongs, as well as the object a2 whose data is to be read. The application program 102 processes exclusively messages which refer to objects of the class A'. No further measures are taken in the application program 102 with respect to  
25 the class A.

The database program 110 accesses the memory 42 during the processing of the message N1 in order to read the data of the subscriber Tln2 which is stored in the  
30 object a2. The object a2 also contains an origin identifier oC in which the class which has been specified when the object a2 was generated is specified. The database program 110 reads the data requested with the instruction B1 in the object a2 and  
35 enters this data into the result message EN1. In addition, in the result message EN1 there is a note, by means of a response identifier AB1, that the result message EN1 contains the result which has been generated when the instruction B1 was processed.



transferred from the database program 110 in the result message EN1 to the application program 102.

When the application program 102 is processed it is a prerequisite that all the messages to be processed refer to the class A' and not to the class A. The value of the origin identifier oC=A is transferred as a value of the class identifier moC in the confirmation message BN1 by the application program 102. For this value assignment it is not necessary for the application program 102 to be able to process objects of the class A. In addition, the confirmation message BN1 contains the origin identifier oC=A and the interrogated subscriber data of the subscriber Tln2. The interface program 100 does not need to remember the class identifier of the maintenance message WN1 by virtue of this procedure.

The confirmation message BN1 is processed by the interface program 100 and transmitted, as confirmation message BN1', to the service computer 24 on the transmission link 22 in accordance with the transmission protocol. By reference to the table T, the interface program 100 determines which data fields in the message BN1 are not contained in objects of the class A. These data fields are not transferred into the confirmation message BN1'.

The service computer 24 receives the confirmation message BN1' and can process it as a message which refers to an object of the class A. Objects of the class A' in the control computer 36 are controlled from the service computer 24 as if they were objects of the class A. The development in the control computer 36 does not therefore restrict the operating features of the service computer 24.

Figure 4 shows the processing of messages in the control computer 16 after the development of the class A into the class A', the object a3 which is not generated until after the development being controlled.

099544-021303



By means of inputs by an operator into the service computer 24, a maintenance message WN2 is generated in the service computer 24, said maintenance message WN2 having the function of generating in the control  
5 computer 16 the object a3 for the subscriber data of the subscriber Tln3. For this reason, the maintenance message WN2 contains a coding for the instruction "Generate" in an instruction field. The class identifier moC in the maintenance message WN2  
10 identifies the class A as the class to which the maintenance message WN2 refers. The object identifier moI of the maintenance message WN2 identifies the object a3 which is to be generated via the maintenance message WN2. The maintenance message WN2 also contains  
15 further data which is not illustrated.

The interface program 100 processes the maintenance message WN2 in the same way as explained above for the maintenance message WN1. When the maintenance message  
20 WN2 is processed, the table T stored in the memory 122 is again used to determine the alternative class A' for the class A specified in the class identifier moC of the maintenance message WN2 and to use it as the class identifier moC of the maintenance message WN2'. The  
25 object identifier moI=a3 has the same value in the maintenance message WN2' as in the maintenance message WN2. The remaining data is also transferred from the maintenance message WN2 into the maintenance message WN2' when the interface program 100 is executed. In  
30 addition, in the identifier allo of the maintenance message WN2', it is noted that the class A is the class with respect to which the class A' is allomorphous. The maintenance message WN2' generated in this way is transmitted from the interface program 100 to the  
35 application program 102 in accordance with the protocol in the control computer 36.

When the maintenance message WN2' is processed, the application program 102 transmits a message N2 to the

Variable	Mean	Standard deviation	Minimum	Maximum
Age	34.5	10.5	20	55
Gender	0.5	0.5	0	1
Marital status	0.5	0.5	0	1
Education	12.5	1.5	10	15
Income	1.5	0.5	1	2
Health status	0.5	0.5	0	1
Smoking status	0.5	0.5	0	1
Alcohol consumption	0.5	0.5	0	1
Exercise frequency	0.5	0.5	0	1
Stress level	0.5	0.5	0	1
Sleep quality	0.5	0.5	0	1
Work satisfaction	0.5	0.5	0	1
Life satisfaction	0.5	0.5	0	1
Overall health	0.5	0.5	0	1
Physical activity	0.5	0.5	0	1
Mental health	0.5	0.5	0	1
Social support	0.5	0.5	0	1
Work-life balance	0.5	0.5	0	1
Financial stability	0.5	0.5	0	1
Family harmony	0.5	0.5	0	1
Personal growth	0.5	0.5	0	1
Community involvement	0.5	0.5	0	1
Environmental awareness	0.5	0.5	0	1
Cultural appreciation	0.5	0.5	0	1
Artistic expression	0.5	0.5	0	1
Volunteer work	0.5	0.5	0	1
Charitable contributions	0.5	0.5	0	1
Philanthropic activities	0.5	0.5	0	1
Leadership roles	0.5	0.5	0	1
Networking opportunities	0.5	0.5	0	1
Professional development	0.5	0.5	0	1
Continuous learning	0.5	0.5	0	1
Adaptability to change	0.5	0.5	0	1
Resilience to stress	0.5	0.5	0	1
Emotional regulation	0.5	0.5	0	1
Interpersonal skills	0.5	0.5	0	1
Conflict resolution	0.5	0.5	0	1
Decision-making skills	0.5	0.5	0	1
Problem-solving abilities	0.5	0.5	0	1
Time management	0.5	0.5	0	1
Organization skills	0.5	0.5	0	1
Communication skills	0.5	0.5	0	1
Teamwork abilities	0.5	0.5	0	1
Leadership qualities	0.5	0.5	0	1
Networking skills	0.5	0.5	0	1
Professional growth	0.5	0.5	0	1
Continuous improvement	0.5	0.5	0	1
Adaptability to change	0.5	0.5	0	1
Resilience to stress	0.5	0.5	0	1
Emotional regulation	0.5	0.5	0	1
Interpersonal skills	0.5	0.5	0	1
Conflict resolution	0.5	0.5	0	1
Decision-making skills	0.5	0.5	0	1
Problem-solving abilities	0.5	0.5	0	1
Time management	0.5	0.5	0	1
Organization skills	0.5	0.5	0	1
Communication skills	0.5	0.5	0	1
Teamwork abilities	0.5	0.5	0	1
Leadership qualities	0.5	0.5	0	1
Networking skills	0.5	0.5	0	1
Professional growth	0.5	0.5	0	1
Continuous improvement	0.5	0.5	0	1
Adaptability to change	0.5	0.5	0	1
Resilience to stress	0.5	0.5	0	1
Emotional regulation	0.5	0.5	0	1
Interpersonal skills	0.5	0.5	0	1
Conflict resolution	0.5	0.5	0	1
Decision-making skills	0.5	0.5	0	1
Problem-solving abilities	0.5	0.5	0	1
Time management	0.5	0.5	0	1
Organization skills	0.5	0.5	0	1
Communication skills	0.5	0.5	0	1
Teamwork abilities	0.5	0.5	0	1
Leadership qualities	0.5	0.5	0	1
Networking skills	0.5	0.5	0	1
Professional growth	0.5	0.5	0	1
Continuous improvement	0.5	0.5	0	1
Adaptability to change	0.5	0.5	0	1
Resilience to stress	0.5	0.5	0	1
Emotional regulation	0.5	0.5	0	1
Interpersonal skills	0.5	0.5	0	1
Conflict resolution	0.5	0.5	0	1
Decision-making skills	0.5	0.5	0	1
Problem-solving abilities	0.5	0.5	0	1
Time management	0.5	0.5		

the expanded class A' and the name of the object a3 to be created.

When the message N2 is processed, the object a3 is  
5 generated in the memory 42 by means of the database  
program 110 by assigning to this object specific memory  
cells which are filled with initial values. In the  
origin identifier oC of the object a3 there is a note  
of the class A' because this class was specified when  
10 the object a3' was generated. An identifier allo in the  
object a3 refers to the class A because the object a3  
is allomorphic with respect to the class A.

The database program 110 generates a result message EN2  
15 in order to confirm the generation of the object a3.  
The result message EN2 contains a response identifier  
AB2 which specifies that the result message EN2 was  
produced when an instruction "Generate" was produced.  
In addition, the result message EN2 contains an  
20 identifier which refers to the class A', the origin  
identifier oC=A', the identifier allo=A and further  
data (not illustrated) of the object a3.

The result message EN2 is processed by the application  
25 program 102, a confirmation message BN2 being  
generated. The value of the origin identifier oC=A' is  
used as a value of the class identifier moC in the  
confirmation message BN2. The other data items in the  
result message EN2 are transferred into the  
30 confirmation message BN2.

When the interface program 100 is processed, after the  
reception of the confirmation message BN2, a  
confirmation message BN2' is generated in accordance  
35 with the transmission protocol used on the transmission  
link 22. The confirmation message BN2' contains all the  
data of the confirmation message BN2 because the  
interface program 100 determines, by reference to the  
table T, that it is not necessary to remove any data

00000000-00000000



During the processing of the confirmation message BN2' in the service computer 24, it is determined by reference to the identifier allo that the confirmation message BN2' relates to data of an object of the class A or that the confirmation message BN2' can be treated as if it contained data of an object of the class A. Although the program in the service computer 24 has not been changed after development in the control computer 36, objects of the class A' in the control computer 36 can be controlled from the control computer 24 as if they were objects of the class A. As long as only the class A is known in the service computer A, it is also the case that only the data items contained in objects of the class A are processed in the confirmation message BN2'.

By means of the method explained with reference to figures 3 and 4, it is also possible to support classes which are allomorphic with respect to a plurality of classes. If, for example, the class A' is developed into a class A'', objects can be allomorphic with respect to the classes A' and A. The control computer 36 can then be executed by service computers 24 during whose programming at least one of the classes A, A' or A'' was known. It is possible to dispense with the class A as soon as all the service computers know at least the class A'.

The method explained with reference to figures 3 and 4 is applied in maintenance messages for generating objects, and in maintenance messages for reading objects. In addition, this method can be used to change data in objects and delete objects. For these methods to operate without errors, the service computer 24 must be capable:

- of assigning the confirmation messages BN to the associated maintenance messages WN,

- of reading and evaluating the identifier allo in the confirmation messages BN,
- 5 - of skipping over unknown name associations (cf. explanations relating to fig. 5) and unknown identifier values and parameters,
- of storing the origin class of an object if there is a database in the service computer,
- 10 - of skipping over unknown optional values, and
- of skipping over unknown values of the "enumerated" data type.

15

In other words, the service computer 24 must be programmed in such a way that objects can be transferred from it into a control computer with a greater degree of knowledge.

20

The interface program 100 has the following features:

- When a class is expanded it is necessary to incorporate the expanded class into the table T,
- 25 which means that for each expanded class it is necessary to store the classes with respect to which the expanded class is allomorphous.
- The name associations which have been made by the
- 30 expanded class must be stored in the table T.
- The interface program 100 is to remove from the confirmation messages BN coming from the application program 102 the data which does not
- 35 belong to the compatible class if the confirmation message BN transmitted to the service computer 24 does not relate to the non-expanded class.

- The interface program 100 must be able to process filter instructions. This is explained in more detail below with reference to figure 5.

5 The application program 102 fulfills the following requirements:

- ```

- As soon as the system moves over to the expanded
  class, only objects of the new class are then
10 generated, irrespective of the knowledge of the
   service computer.

- The new classes are either expanded classes or
  classes which have nothing to do with the previous
15 classes.
```

The database program 110 is configured in such a way that, after a development, the entire data stock relating to the origin class is converted into a data  
20 stock of the expanded class.

The creation of the table T and the conversion of the data stock in the database of the database program 110 are supported by service programs. These service programs evaluate description languages which are used to specify the expansion of classes.

Conditions which have to be complied with when taking into consideration allomorphy are dealt with below.

30 These conditions apply on the level of the classes although, according to standard X.720, allomorphy is firstly a property of an object. In the standard X.720, the following conditions are mentioned:

- 35 - conditions for the expanded class in  
section 5.2.2.1,

- conditions for what are referred to as program packages in the sections 5.2.2.1 and 5.2.2.2,
- 5    - conditions for identifier values in the sections 5.2.2.3 and 5.3.4.1,
- conditions for what are referred to as identifier groups in section 5.2.2.4,
- 10   - conditions for actions, confirmations and parameters in section 5.2.2.54,
- conditions for the behavior of the objects in section 5.2.2.6, and
- 15   - conditions for the name association in section 5.3.4.1.

Figure 5 shows, in a part a, a section of what is referred to as a binary tree which defines the name association of objects of the class A and A'. Name association is the assignment of the object to what is referred to as a superordinate object, said assignment being used for the definition of a uniquely defined name for an object. Identical object names for different objects can be used if the objects each belong to another superordinate object. A name which is uniquely defined in the control computer 16 is then produced from the name of the superordinate object and the name of the objects which are subordinated in this way. The superordinate object is to be specified when an object is generated. The subordinate object is considered as being contained in the superordinate object, referred to in English as "containment".

35

An object b1 of the class B is the superordinate object for the objects a2 and a3, which both belong to the class A' after the development, but have different original classes oC=A and oC=A'. The name associations



[illegible]

For the portion of the name tree shown in part A of figure 5 there is a note in table T that objects of the class B contain subordinate objects of the original class A and of the class A'. At each development, the  
5 name associations in the table T are also adapted to the name tree which is also decisive after the development.

In part b of figure 5, an access to objects of both  
10 classes A and A' is illustrated after the service computer 24 also knows both classes A and A'. A maintenance message WN3 contains a filter instruction F1=((oC=A) OR (oC=A')) in which it is defined that subordinate objects of the class A or A' are to be  
15 registered. The class identifier moC=B of the maintenance message WN3 specifies that the maintenance message WN3 also refers to the class B. The object identifier moI=b1 of the maintenance message WN3 specifies that the object b1 is to be processed, i.e.  
20 is the superordinate object.

When the interface program 100 is executed in the switching office 16, the filter instruction F1 is transmitted unchanged to the application program 102.  
25 This means in particular that operations which refer to the original class A do not have to be replaced in the filter instruction F1 by operations which refer to the expanded class A'. This measure ensures that the service computer 24 can distinguish between objects of  
30 the original class A and of the original class A'.

The application program 102 causes the data of the objects a2 and a3 whose original class is the class A and A', respectively, to be read from the memory 42 by  
35 means of database program 110. Two confirmation messages BN3 and BN4 (not illustrated) are transmitted to the interface program 100. A new confirmation message BN3', which contains the class identifier moC=A, the object identifier moI=a2,

5

- 15

20

Patent claims

1. A method for operating a telecommunications network (10, 12), in which a network element (16) at a network node of a telecommunications network (12) is controlled by a control computer (36), at least one interface program (100) and a plurality of application programs (102, 104) are stored in the control computer (36) in addition to the operating system, during the execution of which application programs (102, 104) application objects (a2, a3) are processed, the application objects (a2, a3) having, depending on their membership of a class (A, A'), data with a predefined data structure and preferably also predefined methods for processing the data, a link via which maintenance is performed on the control computer (36) by means of at least one maintenance message (WN1 to WN3) is set up between a service computer (24) and the control computer (36), the interface program (100) processing the maintenance messages (WN1 to WN3) coming from the service computer (24), the maintenance message (WN1 to WN3) contains a class identifier (moC) which assigns the maintenance message (WN1 to WN3) to a class (A, A'), the class identifier (moC) of the maintenance message (WN1 to WN3) specifies the class (A), known in the service computer, of an application object (a2, a3) to be processed, when the interface program (100) is processed with reference to the class identifier (moC) an alternative identifier is determined which indicates an alternative class (A') to which the application object (a2, a3) to be processed belongs in the network element (16), when the interface program (100) is processed the

03.12.2001  
199901391

- 24a -

EP 000002070

alternative identifier (A') is incorporated into  
an amended maintenance message (WN1' to WN2'),

AMENDED SHEET

and when the amended maintenance message (WN1' to WN2') is processed, the application object (a2, a3) to be processed is processed as an object of the alternative class (A') by an application program (102).

2. The method as claimed in one of the preceding claims, characterized in that, when the alternative identifier (A') is determined, a first table (T) which is stored in the memory (122) of the control computer (16) is used, in which table (T) an alternative identifier (A') is assigned to the class identifier (A).
3. The method as claimed in claim 1 or 2, characterized in that, after the processing of the amended maintenance message (WN1'), the application program (102) generates a confirmation message (BN1) in which the class (oC) specified when the application object (a2, a3) to be processed was generated is specified.
4. The method as claimed in claim 3, characterized in that, when the interface program (100) is processed, an amended confirmation message (BN1') is generated from the confirmation message (BN1), said amended confirmation message (BN1') containing only data which has an application object (a2) of the class (A) to which the confirmation message (BN1) relates, and in that the amended confirmation message (BN1') is transmitted to the service computer (24).
5. The method as claimed in claim 3 or 4, characterized in that the class (A, A') specified when the application object (a2, a3) to be

03.12.2001  
199901391

- 25a -

EP 000002070

5 processed is generated is stored as an origin  
class (oC) in the data of the application object  
(a2, a3) to be processed, and in that, when the  
application program (102) is processed, the origin  
class (oC) is used as a class identifier (moC).

AMENDED SHEET

6. The method as claimed in one of claims 3 to 5, characterized in that the confirmation message (BN2) contains an identifier (allo) in which at least one class (A) is designated which is known in the service computer (24) as the class (A) to which the application object (a2, a3) to be processed belongs.
7. The method as claimed in claim 6, characterized in that the confirmation message (BN1) contains, in addition to the class identifier (moC), an origin identifier (oC) in which the origin class (A, A') is specified.
8. The method as claimed in claim 6 or 7, characterized in that at least one class (A) known in the service computer (24) and/or in at least one other service computer for the application object is stored as an allomorph class (allo) in the data of the application object (a2, a3), and in that, when the application program (102) is processed, the allomorph class (allo) is used as an auxiliary identifier (allo).
9. The method as claimed in one of claims 4 to 8, characterized in that the or a confirmation message (BN1', BN2') which is generated for the service computer (24) when the interface program (100) is being processed contains the class identifier (moC) and the origin identifier (oC) or the class identifier (moC), the origin identifier (oC) and an identifier (allo) in which at least one class (A) is designated.
10. The method as claimed in one of the preceding claims, characterized in that the network element



03.12.2001  
199901391

- 26a -

EP 000002070

is a switching office (16), a cross-connector or a concentrator unit.

- 5 11. The method as claimed in one of the preceding claims, characterized in that the telecommunications network (12)

is a fixed network, a mobile radio network or a network with a fixed network component and a mobile radio network component.

- 5 12. The method as claimed in one of the preceding claims, characterized in that the interface program (100) carries out further interface functions between the service computer (24) and the application programs (102, 104),  
10 preferably an event control for defining the processing sequence of the maintenance messages (WN1 to WN3) and/or an adaptation of the messages (WN1 to WN3) coming from the service computer (24) to a protocol for transmitting messages within the control computer (16), and/or an adaptation of the  
15 confirmation messages (BN1, BN2) coming from the application programs (102, 104) to a predefined protocol for transmitting messages between the service computer (24) and control computer (16).
- 20 13. The method as claimed in one of the preceding claims, characterized in that a first application program (102) is used for subscriber administration,  
25 and/or a second application program is used for administering connecting lines to other switching devices,  
and/or a third application program is used for performing maintenance on the switching devices,  
30 and/or a fourth application program (104) is used for measuring the traffic on the switched links (18).
- 35 14. The method as claimed in claim 13, characterized in that the application objects (a2, a3) of the first application program (102) contain the

subscriber data, preferably the call number and/or the useable telecommunications services, for one subscriber (Tln2, Tln3) in each case.

- 5 15. The method as claimed in one of the preceding claims, characterized in that the maintenance messages (WN1 to WN3) contain a name identifier (moI) for the name of the application

object (a2, a3) to which the maintenance message (WN1 to WN3) relates.

- 5 16. A network element (16) for operating a telecommunications network (10, 12), characterized in that it has a control computer (36) which has an interface program (100) and a plurality of application programs (102, 104), and in that the control computer (36) carries out the method as claimed in one of the preceding claims by processing the programs (100, 102, 104).
- 10
- 15 17. A telecommunications network (10, 12) characterized in that it contains a multiplicity of network elements, and in that it connects the subscribers which are connected to the network elements, the telecommunications network (10, 12) containing at least one network element (16) as claimed in claim 16.

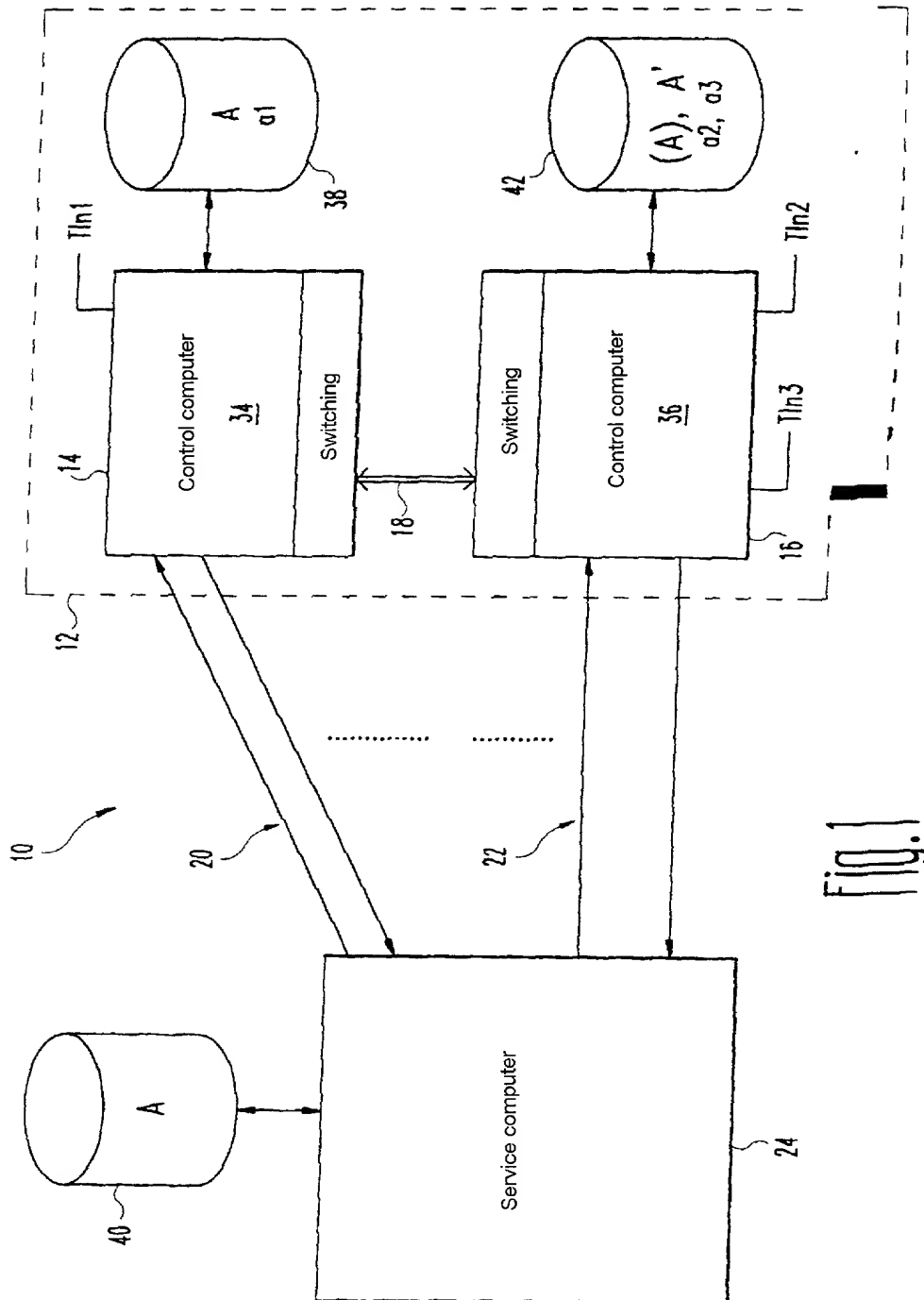


Fig. 1

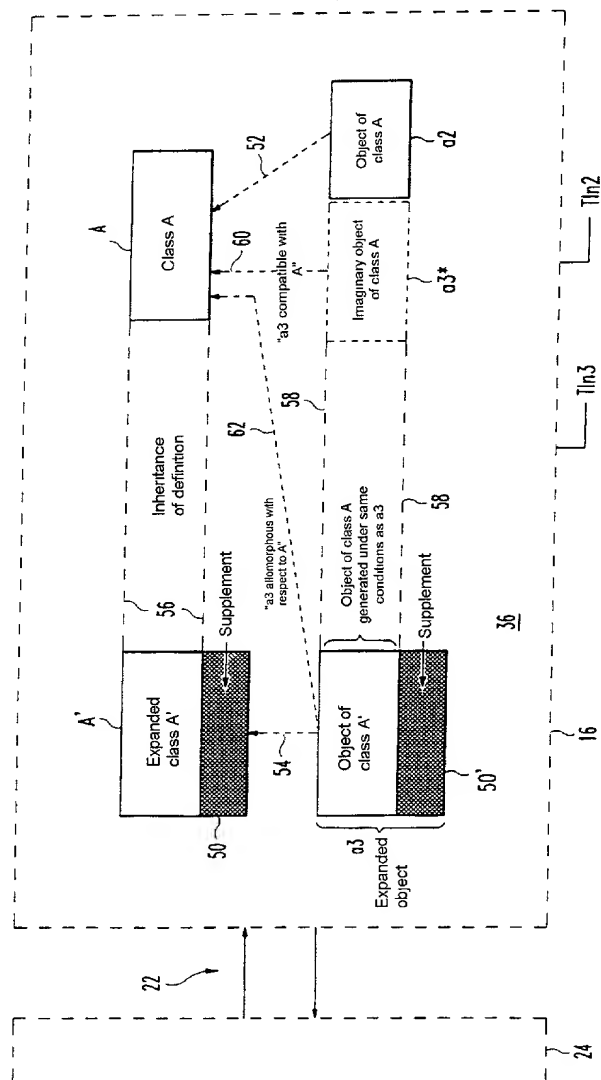
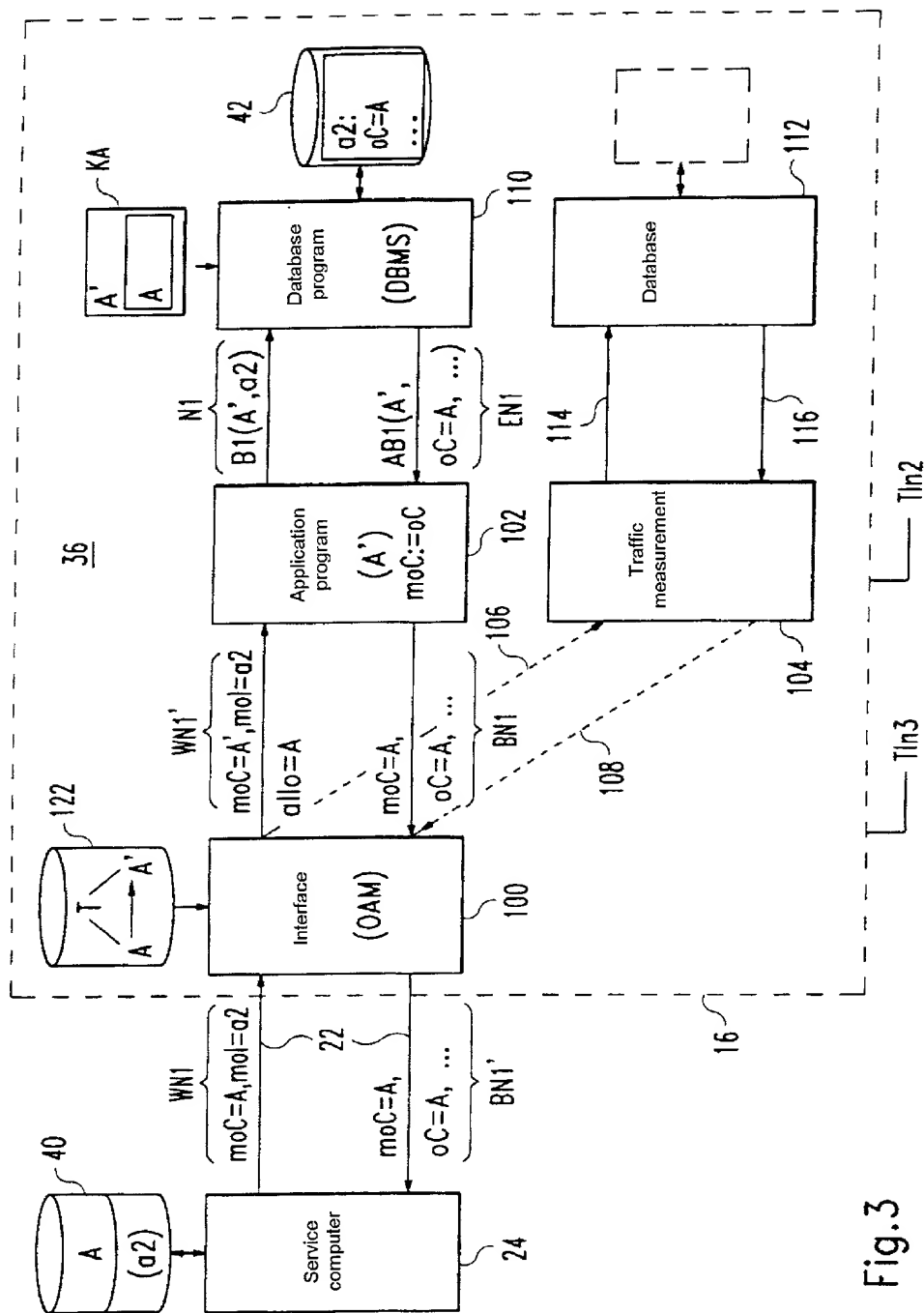


Fig. 2



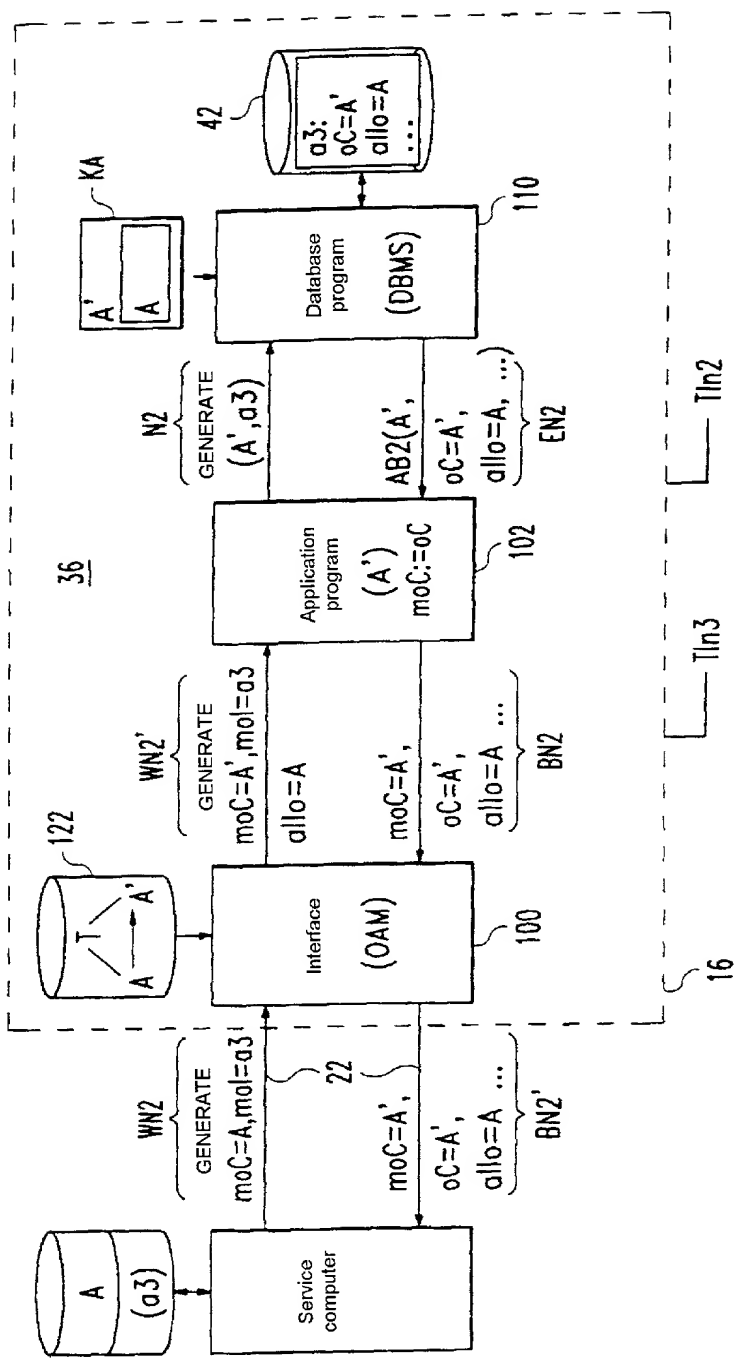


Fig.4



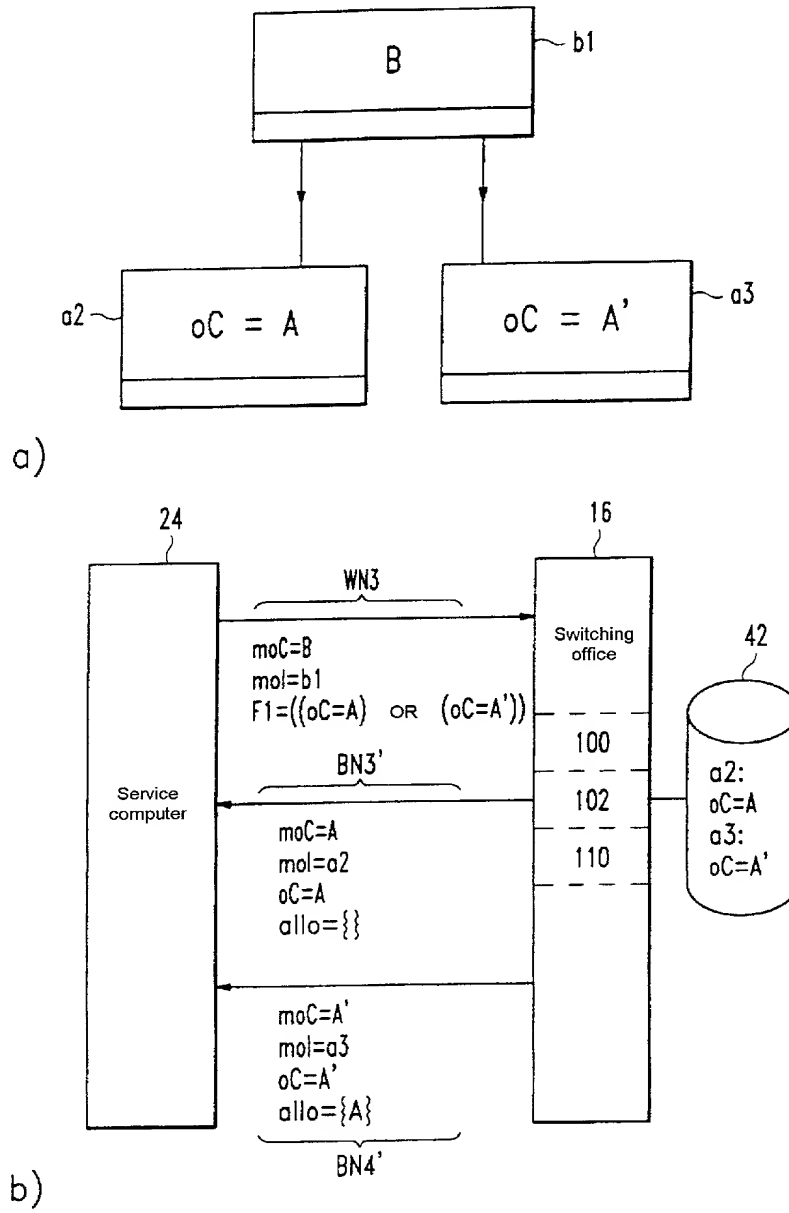


Fig.5

# Declaration and Power of Attorney For Patent Application

## Erklärung Für Patentanmeldungen Mit Vollmacht

### German Language Declaration

Als nachstehend benannter Erfinder erkläre ich hiermit an Eides Statt:

As a below named inventor, I hereby declare that:

dass mein Wohnsitz, meine Postanschrift, und meine Staatsangehörigkeit den im Nachstehenden nach meinem Namen aufgeführten Angaben entsprechen,

My residence, post office address and citizenship are as stated below next to my name,

dass ich, nach bestem Wissen der ursprüngliche, erste und alleinige Erfinder (falls nachstehend nur ein Name angegeben ist) oder ein ursprünglicher, erster und Miterfinder (falls nachstehend mehrere Namen aufgeführt sind) des Gegenstandes bin, für den dieser Antrag gestellt wird und für den ein Patent beantragt wird für die Erfindung mit dem Titel:

I believe I am the original, first and sole inventor (if only one name is listed below) or an original, first and joint inventor (if plural names are listed below) of the subject matter which is claimed and for which a patent is sought on the invention entitled

Verfahren und Netzelement zum  
Betreiben eines  
Telekommunikationsnetzes

Method and network element for  
operating a telecommunications network

deren Beschreibung

the specification of which

(zutreffendes ankreuzen)

☐ hier beigefügt ist.

☒ am 09.03.2000 als

PCT internationale Anmeldung

PCT Anwendungsnummer PCT/EP00/02070

eingereicht wurde und am \_\_\_\_\_  
abgeändert wurde (falls tatsächlich abgeändert).

(check one)

☐ is attached hereto.

☒ was filed on 09.03.2000 as

PCT international application

PCT Application No. PCT/EP00/02070

and was amended on \_\_\_\_\_  
(if applicable)

Ich bestätige hiermit, dass ich den Inhalt der obigen Patentanmeldung einschliesslich der Ansprüche durchgesehen und verstanden habe, die eventuell durch einen Zusatzantrag wie oben erwähnt abgeändert wurde.

I hereby state that I have reviewed and understand the contents of the above identified specification, including the claims as amended by any amendment referred to above.

Ich erkenne meine Pflicht zur Offenbarung irgendwelcher Informationen, die für die Prüfung der vorliegenden Anmeldung in Einklang mit Absatz 37, Bundesgesetzbuch, Paragraph 1.56(a) von Wichtigkeit sind, an.

I acknowledge the duty to disclose information which is material to the examination of this application in accordance with Title 37, Code of Federal Regulations, §1.56(a).

Ich beanspruche hiermit ausländische Prioritätsvorteile gemäss Abschnitt 35 der Zivilprozessordnung der Vereinigten Staaten, Paragraph 119 aller unten angegebenen Auslandsanmeldungen für ein Patent oder eine Erfindersurkunde, und habe auch alle Auslandsanmeldungen für ein Patent oder eine Erfindersurkunde nachstehend gekennzeichnet, die ein Anmeldedatum haben, das vor dem Anmeldedatum der Anmeldung liegt, für die Priorität beansprucht wird.

I hereby claim foreign priority benefits under Title 35, United States Code, §119 of any foreign application(s) for patent or inventor's certificate listed below and have also identified below any foreign application for patent or inventor's certificate having a filing date before that of the application on which priority is claimed:

# German Language Declaration

Prior foreign applications  
Priorität beansprucht

Priority Claimed

99104749.9

EP

10.03.1999

☒

☐

(Number)  
(Nummer)

(Country)  
(Land)

(Day Month Year Filed)  
(Tag Monat Jahr eingereicht)

Yes  
Ja

No  
Nein

(Number)  
(Nummer)

(Country)  
(Land)

(Day Month Year Filed)  
(Tag Monat Jahr eingereicht)

☐  
Yes  
Ja

☐  
No  
Nein

(Number)  
(Nummer)

(Country)  
(Land)

(Day Month Year Filed)  
(Tag Monat Jahr eingereicht)

☐  
Yes  
Ja

☐  
No  
Nein

Ich beanspruche hiermit gemäss Absatz 35 der Zivilprozessordnung der Vereinigten Staaten, Paragraph 120, den Vorzug aller unten aufgeführten Anmeldungen und falls der Gegenstand aus jedem Anspruch dieser Anmeldung nicht in einer früheren amerikanischen Patentanmeldung laut dem ersten Paragraphen des Absatzes 35 der Zivilprozessordnung der Vereinigten Staaten, Paragraph 122 offenbart ist, erkenne ich gemäss Absatz 37, Bundesgesetzbuch, Paragraph 1.56(a) meine Pflicht zur Offenbarung von Informationen an, die zwischen dem Anmeldedatum der früheren Anmeldung und dem nationalen oder PCT internationalen Anmeldedatum dieser Anmeldung bekannt geworden sind.

I hereby claim the benefit under Title 35, United States Code, §120 of any United States application(s) listed below and, insofar as the subject matter of each of the claims of this application is not disclosed in the prior United States application in the manner provided by the first paragraph of Title 35, United States Code, §122, I acknowledge the duty to disclose material information as defined in Title 37, Code of Federal Regulations, §1.56(a) which occurred between the filing date of the prior application and the national or PCT international filing date of this application.

PCT/EP00/02070  
(Application Serial No.)  
(Anmeldeseriennummer)

09.03.2000  
(Filing Date D, M, Y)  
(Anmeldedatum T, M, J)

anhängig  
(Status)  
(patentiert, anhängig,  
aufgegeben)

pending  
(Status)  
(patented, pending,  
abandoned)

(Application Serial No.)  
(Anmeldeseriennummer)

(Filing Date D,M,Y)  
(Anmeldedatum T, M, J)

(Status)  
(patentiert, anhängig,  
aufgeben)

(Status)  
(patented, pending,  
abandoned)

Ich erkläre hiermit, dass alle von mir in der vorliegenden Erklärung gemachten Angaben nach meinem besten Wissen und Gewissen der vollen Wahrheit entsprechen, und dass ich diese eidesstattliche Erklärung in Kenntnis dessen abgebe, dass wissentlich und vorsätzlich falsche Angaben gemäss Paragraph 1001, Absatz 18 der Zivilprozessordnung der Vereinigten Staaten von Amerika mit Geldstrafe belegt und/oder Gefängnis bestraft werden koennen, und dass derartig wissentlich und vorsätzlich falsche Angaben die Gültigkeit der vorliegenden Patentanmeldung oder eines darauf erteilten Patentes gefährden können.

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true, and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application or any patent issued thereon.

RECEIVED 11/15/99

# German Language Declaration

VERTRETUNGSVOLLMACHT: Als benannter Erfinder beauftrage ich hiermit den nachstehend benannten Patentanwalt (oder die nachstehend benannten Patentanwälte) und/oder Patent-Agenten mit der Verfolgung der vorliegenden Patentanmeldung sowie mit der Abwicklung aller damit verbundenen Geschäfte vor dem Patent- und Warenzeichenamt: (Name und Registrationsnummer anführen)

POWER OF ATTORNEY: As a named inventor, I hereby appoint the following attorney(s) and/or agent(s) to prosecute this application and transact all business in the Patent and Trademark Office connected therewith. (list name and registration number)



29177

PATENT TRADEMARK AND I hereby appoint

Customer No.

Telefongespräche bitte richten an:  
(Name und Telefonnummer)

Direct Telephone Calls to: (name and telephone number)

Ext. \_\_\_\_\_

Postanschrift:

Send Correspondence to:

Bell, Boyd & Lloyd LLC  
Three First National Plaza, 70 West Madison Street, Suite 3300 60602-4207 Chicago, Illinois  
Telephone: (001) 312 372 11 21 and Facsimile (001) 312 372 20 98  
or  
Customer No.

|                                                                                         |                          |                                                                            |      |
|-----------------------------------------------------------------------------------------|--------------------------|----------------------------------------------------------------------------|------|
| Voller Name des einzigen oder ursprünglichen Erfinders:<br><b>MARIA FROESCHL</b>        |                          | Full name of sole or first inventor:<br><b>MARIA FROESCHL</b>              |      |
| Unterschrift des Erfinders<br><i>Maria Froeschl</i>                                     | Datum<br><b>17.10.01</b> | Inventor's signature<br><i>DEX</i>                                         | Date |
| Wohnsitz<br><b>OTTOBRUNN, DEUTSCHLAND</b>                                               |                          | Residence<br><b>OTTOBRUNN, GERMANY</b>                                     |      |
| Staatsangehörigkeit<br><b>DE</b>                                                        |                          | Citizenship<br><b>DE</b>                                                   |      |
| Postanschrift<br><b>UHLANDSTR.10A</b>                                                   |                          | Post Office Address<br><b>UHLANDSTR.10A</b>                                |      |
| <b>85521 OTTOBRUNN</b>                                                                  |                          | <b>85521 OTTOBRUNN</b>                                                     |      |
| Voller Name des zweiten Miterfinders (falls zutreffend):<br><b>Dr. MATTHIAS NEUHAUS</b> |                          | Full name of second joint inventor, if any:<br><b>Dr. MATTHIAS NEUHAUS</b> |      |
| Unterschrift des Erfinders<br><i>Dr. Matthias Neuhaus</i>                               | Datum<br><b>18.10.01</b> | Second Inventor's signature<br><i>DEX</i>                                  | Date |
| Wohnsitz<br><b>MUENCHEN, DEUTSCHLAND</b>                                                |                          | Residence<br><b>MUENCHEN, GERMANY</b>                                      |      |
| Staatsangehörigkeit<br><b>DE</b>                                                        |                          | Citizenship<br><b>DE</b>                                                   |      |
| Postanschrift<br><b>TRIVASTRASSE 15 C</b>                                               |                          | Post Office Address<br><b>TRIVASTRASSE 15 C</b>                            |      |
| <b>80637 MUENCHEN</b>                                                                   |                          | <b>80637 MUENCHEN</b>                                                      |      |

(Bitte entsprechende Informationen und Unterschriften im Falle von dritten und weiteren Miterfindern angeben).

(Supply similar information and signature for third and subsequent joint inventors).